

Лекція 9

1. ПЕРИФЕРІЙНІ ПРИСТРОЇ ЗАГАЛЬНОГО ПРИЗНАЧЕННЯ

1. Загальні положення

Ця лекція присвячена периферійним пристроям, які є в складі мікроконтролерів сімейства. Процес конфігурації кожною пристроєм і його використання демонструється на прикладах. Після того, як розберемося з роботою периферійних пристроїв, тексти прикладів можна буде використовувати в якості основи для написання власних драйверів низького рівня.

2. Порти введення/виведення загального призначення

2.1 Загальні відомості

Мікроконтролери LPC23xx можуть мати до п'яти портів введення/виведення загального призначення GPIO (General Purpose Input Output), кожен з яких містить 32 лінії введення/виведення. Таким чином, в нашому розпорядженні є до 160 виводів. Щоб зберегти сумісність з моделями LPC21xx попереднього сімейства, для керування портами PORT0 та PORT1 використовуються звичайні керуючі регістри, які підключено до шини APB. Однак при використанні цих регістрів зміна стану портів відбувається досить повільно. Тому в мікроконтролерах LPC23xx з'явився другий набір швидких регістрів керування портами введення/виведення (Fast GPIO) (рисунок 1), який розміщено на локальній шині.

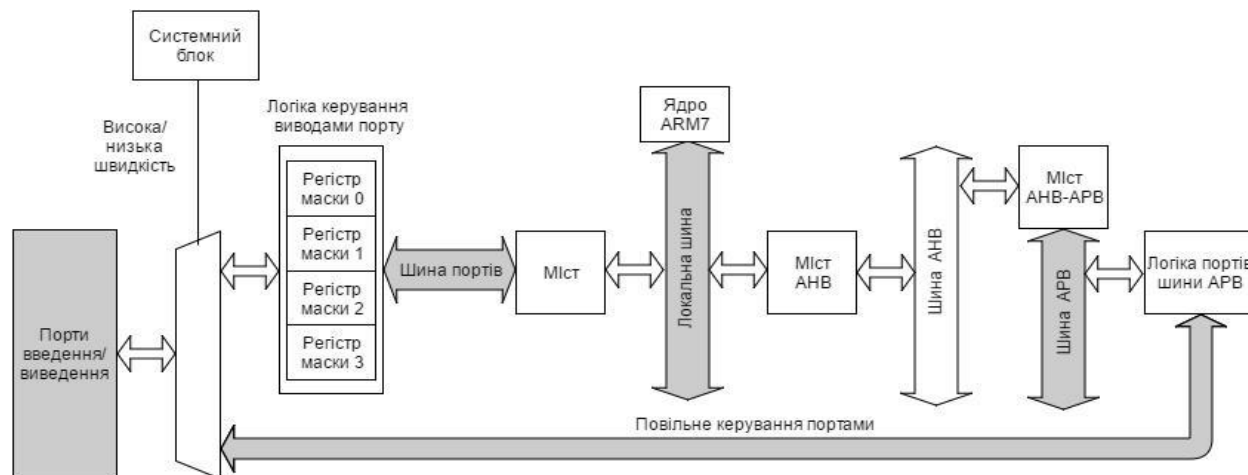


Рисунок 1 – Швидкі регістри GPIO

Рекомендується використовувати саме ці регістри та ігнорувати успадковані регістри GPIO у всіх випадках, за винятком налаштування портів існуючого коду. Окрім цього, порти PORT0 та PORT1 можуть генерувати переривання при появі на заданому виводі контролера зростаючого чи спадаючого фронту сигналу.

Контакти пристрою, які не підключені до конкретної периферійної функції контролюються регістрами введення/виведення загального призначення GPIO. Контакти можуть бути динамічно сконфігуровані як входи або виходи. Окремі регістри дозволяють встановлювати або скидати будь-яку кількість виходів одночасно. Мікроконтролери сімейства LPC2377/78 використовують прискорені функції GPIO:

- регістри GPIO переміщують до локальної шини ARM, що забезпечує досить швидке введення/виведення;
- регістри-маски дозволяють обробляти набори бітів порту у якості групи, залишаючи інші біти без змін;
- усі регістри GPIO адресуються напівсловно та побайтово;
- повне значення порту можна записати в одній інструкції.

Окрім цього, будь-який контакт порту P0 або порту P2 може бути запрограмований для генерації переривання за переднім, заднім фронтом або одним з них. Визначення перепаду асинхронне, тому він може працювати, коли зупинений тактовий генератор, наприклад у режимі зниженого енергоспоживання. Включення переривання може бути використано для пробудження чіпу з режиму Power-Down.

Для ліній введення/виведення загального призначення (GPIO) мікроконтролерів сімейства LPC2300 характерні наступні особливості:

- керування напрямом (вхід це чи вихід) кожної лінії індивідуально;
- розділене керування встановленням та скиданням вихідного рівня лінії;
- усі лінії, за замовчуванням, після скидання є лініями GPIO, які налаштовано як входи.

Лінії GPIO можуть використовуватися для наступних додатків:

- введення/виведення загального призначення;
- керування світлодіодами чи іншими індикаторами;
- керування зовнішніми пристроями, які підключено до мікроконтролера;

– опитування цифрових входів.

2.2 Швидкі регістри введення/виведення

У мікроконтролерах попереднього сімейства LPC2100 регістри керування GPIO розміщувались на шині APB разом з регістрами усіх інших периферійних пристроїв. Однак, при роботі з цими регістрами дуже багато часу витрачалося на пересилання даних по шинам АНВ та APB. Для зміни стану виводу було потрібно 14 тактів, тому під час роботи на частоті 60 МГц максимальна частота переключення виводів порту складала всього 4,3 МГц. В новіших версіях пристроїв LPC213x компанія NXP додала нову групу регістрів, які підключено до локальної шини ARM7 (рисунок 1). Ці регістри дозволяють перемикає вивід порту всього за два такти (максимальна частота перемикаєння – 30 МГц). Окрім швидких регістрів GPIO, було додано ще регістр маски, що спрощує маніпулювання бітами кожного порту.

У сімействі LPC2300 керуючі регістри портів введення/виведення підключені до локальної шини. Таке рішення дозволило значно прискорити керування виводами портів. З міркувань сумісності залишені також «класичні» регістри, за допомогою яких можна керувати портами PORT0 і PORT1.

При скиданні блок керування виводами конфігурує усі виводи периферійних пристроїв як входи портів введення/виведення загального призначення (GPIO). Керування виводами GPIO здійснюється чотирма регістрами (рисунок 2).

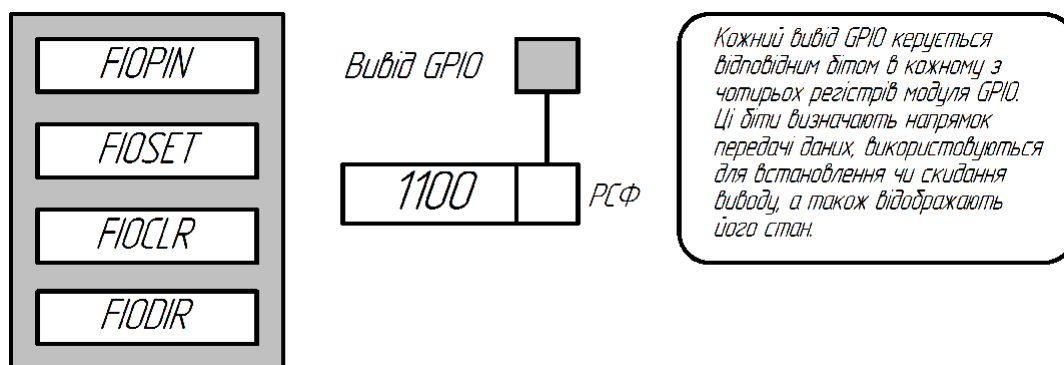


Рисунок 2 – Керування виводами GPIO

Біти регістра FIODIR дозволяють індивідуально конфігурувати кожний вивід у якості входу (0) чи виходу (1). Якщо вивід є виходом, його станом можна керувати за допомогою регістрів FIOSET та FIOCLR. Запис 1 в біти цих регістрів приводить до встановлення (FIOSET) чи скидання (FIOCLR) відповідного виводу. Стан виводу GPIO можна визначити в будь-який момент. Для цього треба прочитати вміст регістра FIOPIN. Регістр FIOMASK використовується для маскування окремих бітів регістрів FIOSET, FIOCLR та FIOPIN. Якщо біт регістра маски скинути в 0, то відповідні біти регістрів FIOSET, FIOCLR та FIOPIN можуть бути зчитані або доступні для запису. Якщо встановлений біт регістра маски – 1, то запис у відповідні біти регістрів неможливий, а читання з них не відображатиметься у регістрі FIOPIN. Таке маскування допомагає прискорити маніпулювання лініями введення/виведення на низькому рівні.

Нижче наведена програма, яка викликає миготіння світлодіоду.

```
#include <LPC23xx.h>

void main()
{
    int flasher;
    int delay;
    FIO2DIR = 0x000000FF;
    FIO2MASK = 0xFFFFFFFF00;
    while(1)
    {
        FIO2CLR = 0x000000FF;
        for(flasher =1;flasher<0x00000100;flasher = flasher <<1)
        {
            FIO2SET = flasher; //FIO2PIN = flasher;
            for(delay = 0;delay<0xA0000;delay++) //simple delay
loop
                {
                    ;
                }
            }
        }
    }
    //return (0);
}
```

В таблиці 1 наведено опис виводів портів введення/виведення.

Таблиця 1 – Опис виводів портів введення/виведення

Найменування	Тип	Опис
P0.0...P0.31	Введення/виведення	Входи/виходи загального призначення (GPIO)
P1.0...P1.31		
P2.0...P2.31		
P3.0...P3.31		
P4.0...P4.31		

2.3 Опис керуючих регістрів

Для програмування GPIO використовуються 4 регістри (таблиця 2).

Таблиця 2 – Керуючі регістри GPIO

Адреса	Ім'я	Опис виводів	Доступ
0xE0028000	IOPIN	Регістр значення GPIO порту 0. Поточний стан виводів завжди може бути прочитаний через цей регістр незалежно від режиму та напрямку виводу.	R/W
0xE0028004	IOSET	Регістр встановлення GPIO порту 0. Цей регістр керує станом вихідних виводів разом з регістром IOCLR. Запис логічної одиниці у будь-який біт приведе до появи потенціалу високого рівня на відповідному виводі. Запис логічного нуля ефекту не має.	R/W
0xE0028008	IODIR	Регістр напрямку GPIO порту 0. Цей регістр задає напрямки роботи (введення чи виведення) індивідуально для кожного виводу.	R/W
0xE002800C	IOCLR	Регістр скидання GPIO порту 0. Цей регістр керує станом вихідних виводів разом з регістром IOSET. Запис логічної одиниці у будь-який біт приведе до появи потенціалу низького рівня на відповідному виводі та скидання цього біта в регістрі IOSET. Запис логічного нуля ефекту не має.	W

2.4 Регістр значення виводів GPIO

Цей регістр відображає значення сигналів, які подано на вхідні виводи. Значення регістра відображає будь-які зміни зовнішніх сигналів, але тільки на тих виводах мікроконтролера, які були сконфігуровані як лінії GPIO. Для виводів портів, які сконфігуровано не як лінії GPIO (а, наприклад, для виконання інших функцій), контроль з використанням регістра IOPIN не дасть достовірного результату.

Вибір для виводу порту будь-якої однієї функції повністю виключає всі інші функції, які можуть бути доступні на тому ж самому виводі. Часткове виключення з цього правила мають лише входи АЦП. Незалежно від функції, яка вибрана для виводу порту, до якого підключений вхід АЦП, цей вхід АЦП у будь-який момент може бути прочитаний мікроконтролером, та зміна вхідної напруги на ньому буде відображена в результаті аналого-цифрового перетворення.

Однак, коректне читання результату АЦП може бути виконано лише у випадку, коли для даного виводу обрана аналогова функція (при цьому буде активована відповідна схема інтерфейсу між виводом та модулем АЦП). У всіх інших випадках, підключення до даного виводу цифрових логічних схем, необхідних для активізації якоїсь іншої функції, приведе до некоректного проведення аналого-цифрового перетворення.

Зауваження: під час запису у цей регістр, значення передається у вихідний регістр, минаючи IOSET та IOCLR. Потрібно уникати використання цієї особливості через те, що при цьому немає можливості індивідуально керувати виводами.

Таблиця 3 – Опис та адреси регістрів IOPIN (IO0PIN – 0xE0028000, IO1PIN – 0xE0028010, IO2PIN – 0xE0028020, IO3PIN – 0xE0028030)

IOPIN	Опис	Після скидання
31:0	Значення виводів GPIO. Біт 0 відповідає P0.0, ..., біт 31 відповідає P0.31	Не визначено

2.5 Регістр встановлення GPIO

Цей регістр використовується для подання на вихідні виводи GPIO сигналу високого рівня. Запис логічної 1 у відповідний біт приведе до появи на відповідному виводі потенціалу високого рівня. Запис логічного 0 не змінить значення біта та потенціалу на виході. Якщо вивід налаштований як вхід чи виконує альтернативну функцію, то запис 1 в його біт не дасть ніякого ефекту.

Читання з IOSET поверне значення вихідного регістра GPIO, значення у якому сформовано попередніми записами в IOSET та IOCLR (чи IOPIN, що дуже не бажано). Значення цього регістра не відображає стан входів.

Таблиця 4 – Опис та адреси регістрів IOSET (IO0SET – 0xE0028004, IO1SET – 0xE0028014, IO2SET – 0xE0028024, IO3SET – 0xE0028034)

IOSET	Опис	Після скидання
31:0	Значення встановлює біти GPIO. Біт 0 відповідає P0.0, ..., біт 31 відповідає P0.31	0

2.6 Регістр скидання GPIO

Цей регістр призначено для подання на вихідні виводи GPIO сигналу низького рівня. Запис логічної 1 у відповідний біт приведе до появи на відповідному виводі потенціалу низького рівня та очищення цього біта в регістрі IOSET. Запис логічного 0 не змінить значення біта на виході. Якщо вивід налаштований як вхід чи виконує альтернативну функцію, то запис 1 в його біт не дасть ніякого ефекту. Значення цього регістра не відображає стан входів.

Таблиця 5 – Опис та адреси регістрів IOCLR (IO0CLR – 0xE002800C, IO1CLR – 0xE002801C, IO2CLR – 0xE002802C, IO3CLR – 0xE002803C)

IOCLR	Опис	Після скидання
31:0	Значення скидає біти GPIO. Біт 0 відповідає P0.0, ..., біт 31 відповідає P0.31	0

2.7 Регістр напрямку GPIO

Цей регістр використовується для керування напрямком роботи виводів, які використовують GPIO. Значення біта напрямку для того чи іншого виводу задає напрямок передавання даних через цей вивід.

Таблиця 6 – Опис та адреси регістрів IODIR (IO0DIR – 0xE0028008, IO1DIR – 0xE0028018, IO2DIR – 0xE0028028, IO3DIR – 0xE0028038)

IODIR	Опис	Після скидання
31:0	Задає напрямок виводів GPIO (0 – вхід, 1 – вихід). Біт 0 відповідає P0.0, ..., біт 31 відповідає P0.31	0

2.8 Особливості програмування GPIO

Якщо для виводу відповідний біт було встановлено і в регістрі IOSET і в регістрі IOCLR, то рівень на виході встановиться у відповідності з останнім записом. Так, при послідовності:

IO0SET=0x00000080,

IO0CLR=0x00000080,

на виході P0.7 буде низький рівень, тому що запис у IO0CLR був після запису в IO0SET.

Додатки, які потребують одночасної зміни всіх ліній паралельного порту, можуть використовувати прямий доступ до відповідного регістра значень виводів порту GPIO (IOPIN). Припустимо, що якщо виводи P0.8 та P0.15 сконфігуровано як виходи, запис в регістр IO0PIN:

IO0PIN=0x0000C700

дасть той самий ефект, що і послідовність команд

IO0SET=0x0000C700,

IO0CLR=0x00003800.

Варіант з записом в регістри IO0SET та IO0CLR буде виконуватись довше у порівнянні з єдиною командою запису в регістр IO0PIN.

2.9 Переривання від GPIO

Окрім роботи за своїм основним призначенням, порти GPIO0 та GPIO2 можуть генерувати переривання при зміні стану будь-якого з виводів порту.

Два регістри дозволу переривання (на кожний порт) дозволяють генерувати переривання за зростаючим чи спадаючим фронтом індивідуально для кожного з виводів порту (рисунок 3).

Обидві лінії переривань від портів введення/виведення з'єднані в одну лінію EINT3, яка під'єднана до виділеного слоту модуля VIC. При генерації переривання у обробнику потрібно перевірити вміст регістрів статусу, щоб визначити, який з виводів згенерував переривання і, якщо це потрібно, який фронт було зафіксовано – зростаючий чи спадаючий. Як і для інших переривань від периферійних пристроїв, перед виходом з обробника потрібно скинути прапорці переривання.

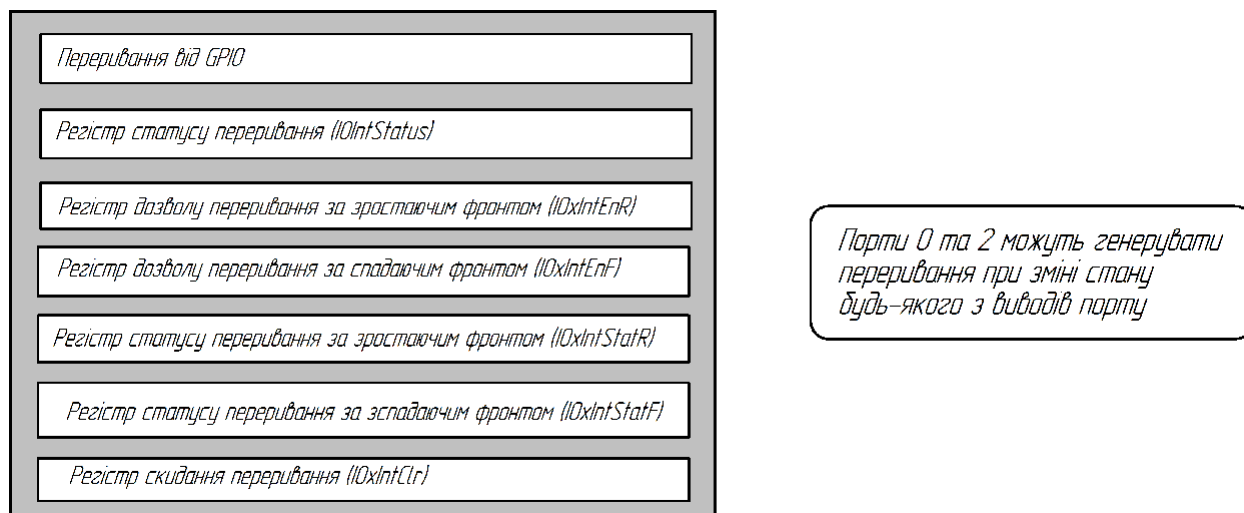


Рисунок 3 – Регістри керування перериваннями GPIO

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) Скільки портів введення/виведення мають мікроконтролери сім'ї LPC23xx?
- 2) Скільки ліній введення/виведення має кожен порт?
- 3) Які керуючі регістри використовуються для керування портами PORT0 та PORT1 та до якої шини їх підключено?
- 4) Який другий набір регістрів керування портами введення/виведення з'явився в мікроконтролерах LPC23xx та на якій шині його розміщено?
- 5) Який фронт сигналу на заданому виводі портів PORT0 та PORT2 може генерувати переривання?
- 6) Які прискорені функції GPIO використовують мікроконтролери сімейства?
- 7) Які особливості характерні для ліній введення/виведення загального призначення (GPIO) мікроконтролерів сімейства LPC2300?
- 8) Наведіть приклади додатків, для яких можуть використовуватися лінії GPIO?
- 9) З якою частотою можуть перемикаєти вивід порту швидкі регістри?
- 10) Назвіть регістри, які можуть використовуватись для керування виводами GPIO.
- 11) Який регістр призначено для програмування кожного виводу у якості входу чи виходу?
- 12) За допомогою яких регістрів можна керувати станом виводу, який є виходом?
- 13) Який регістр призначено для визначення стану виводу?
- 14) Для чого може використовуватися регістр FIOMASK?
- 15) Назвіть регістри, які можуть використовуються для програмування GPIO.
- 16) Дайте опис бітів регістрів IOPIN.
- 17) Дайте опис бітів регістрів встановлення та скидання.
- 18) Дайте опис бітів регістрів напрямку.
- 19) Наведіть особливості програмування GPIO.
- 20) Наведіть особливості генерації переривань від GPIO.

3 Таймери загального призначення

3.1. Загальна характеристика

У складі мікроконтролерів сімейства LPC2300 є чотири таймери загального призначення. Всі ці таймери мають однакову структуру і функціональні можливості. Таймери побудовані на базі 32-бітного лічильника, який об'єднано з 32-бітним переддільником (рисунок 4). В якості тактового сигналу у всіх таймерах використовується тактовий сигнал шини APB (Pclk).

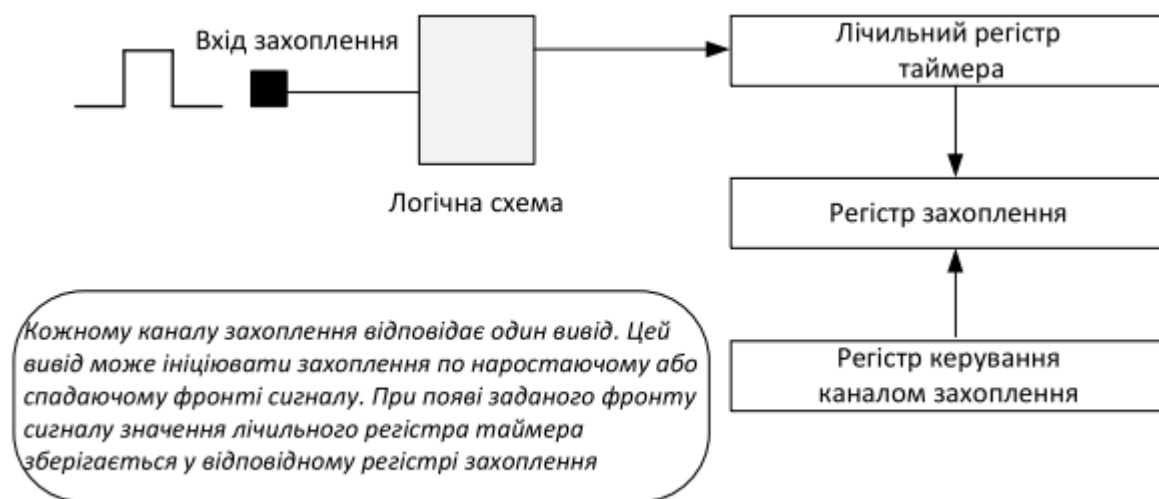


Рисунок 4 – Спрощена структура таймера загального призначення

Швидкість лічби таймера залежить від значення, що знаходиться в реєстрі переддільника. Лічильний реєстр переддільника буде інкрементуватися за кожним імпульсом сигналу Pclk до тих пір, поки не досягне значення, що знаходиться в реєстрі переддільника. У цей момент вміст лічильного реєстра таймера збільшується на 1, лічильний реєстр переддільника скидається в нуль і знову починається лічба. Реєстр керування таймером містить всього два біти, які використовуються для включення/відключення таймера і його скидання.

3.2 Режим захоплення

Поряд з основним лічильником кожен таймер має до чотирьох каналів захоплення (capture). Вони дозволяють запам'ятовувати значення лічильного реєстра таймера за зміною зовнішнього сигналу (рисунок 5).

Кожному каналу захоплення відповідає один з виводів мікроконтролера, використання якого може бути дозволено блоком керування виводів. Вміст реєстра керування каналом захоплення визначає, поява якого із фронтів сигналу (наростаючого, спадаючого або обох) ініціює захоплення.

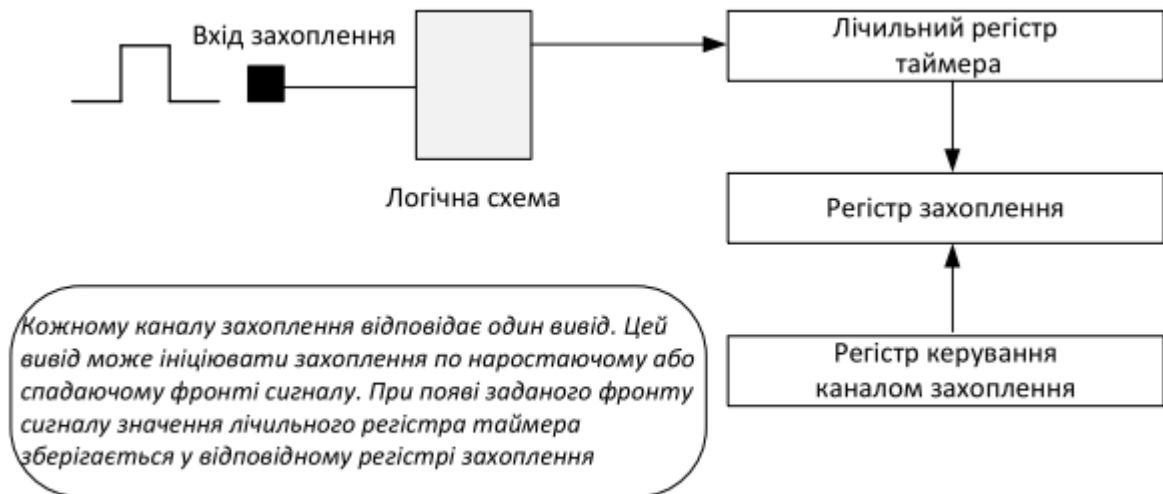


Рисунок 5–Режим захоплення

При настанні цієї події поточне значення лічильного реєстра таймера переписується до відповідного реєстра захоплення і, якщо це необхідно, генерується переривання.

3.3 Режим лічильника зовнішніх подій

Реєстр керування лічбою визначає, яким чином буде використовуватися таймер – як лічильник зовнішніх подій, або як звичайний таймер. За допомогою даного реєстра можна змінити тактовий сигнал таймера з Pclk на сигнал від зовнішнього джерела, що подається на вхід захоплення даного таймера. При цьому лічильний реєстр таймера може інкрементуватися за наростаючим, спадаючим або навіть за обома фронтами зовнішнього сигналу.

3.4 Режим збігу

Кожен таймер має до чотирьох каналів збігу (співпадіння, match). З кожним з цих каналів пов'язаний реєстр збігу, що містить 32-бітне число. Лічильний реєстр таймера безперервно порівнюється з вмістом даного реєстра (рисунок 6). При співпадинні їх значень виникає подія «співпадіння», яка може впливати на таймер (скинути, зупинити або згенерувати переривання), а також

може впливати на стан певного виводу мікроконтролера (встановлення, скидання або перемикання).

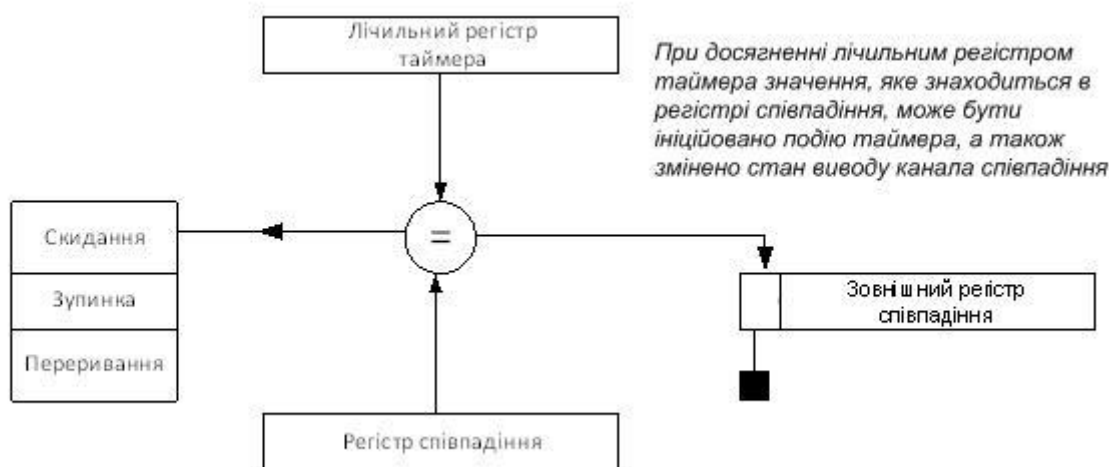


Рисунок 6 – Режим збігу

Щоб скористатися цією можливістю таймера, треба завантажити в регістр співпадіння потрібне значення. Подальша конфігурація блоку співпадіння здійснюється за допомогою регістра керування блоком співпадіння (Match Control Register). У цьому регістрі кожному каналу відповідає набір бітів, що використовуються для дозволу виконання наступних дій за подією збігу: генерація переривання від таймера, скидання таймера, зупинка таймера. Причому допускається будь-яка комбінація зазначених дій. Крім того, кожному каналу співпадіння відповідає певний вивід мікроконтролера, стан якого може змінюватися при співпадінні. Як і в разі каналів захоплення, для використання зазначеної можливості спочатку потрібно за допомогою блоку керування виводами підключити вивід до каналу співпадіння. Стан цих виводів визначається станом чотирьох молодших бітів зовнішнього регістра співпадіння EMR (рисунок 7).

Регістр EMR містить 4-бітове поле, загальне для всіх каналів. Стан бітів цього поля визначає поведінку виводів збігу при виникненні події. Крім того, в регістрі є біти, змінюючи які можна безпосередньо змінювати стан виводу кожного з каналів.

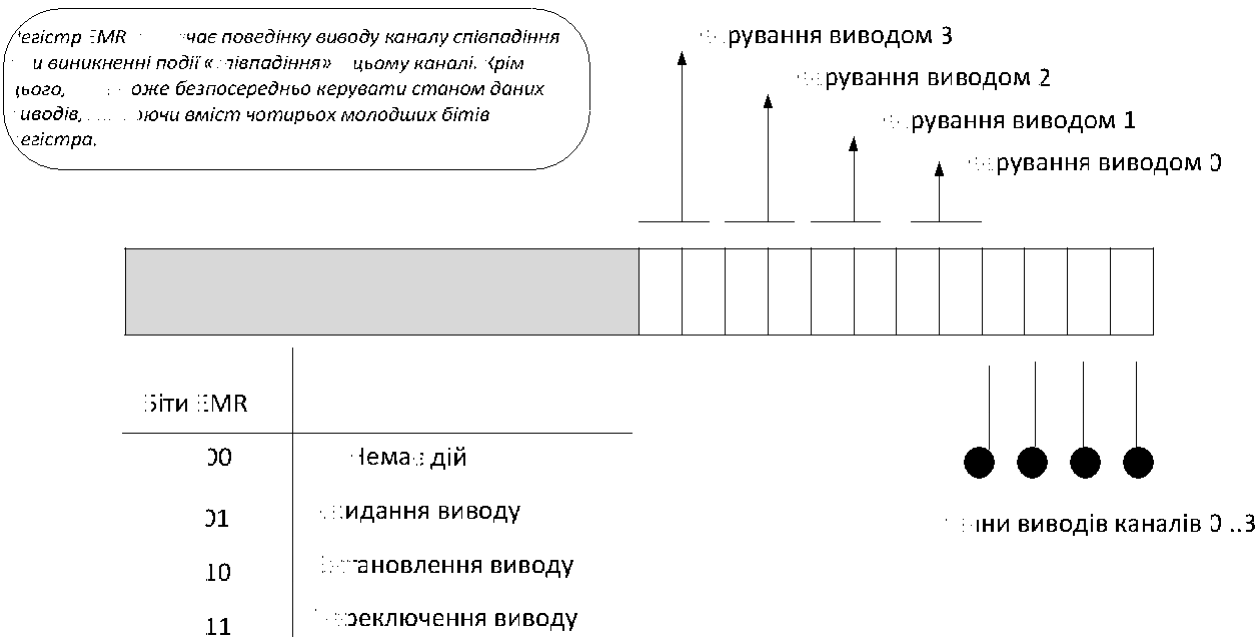


Рисунок 7 – Зовнішній регістр збігу

3.5 Опис керуючих регістрів модуля таймерів

Кожен таймер/лічильник містить керуючі регістри, які показано в таблиці 7 ("значення скидання" відноситься до даних, що зберігаються тільки у використувуваних бітах та не включає зарезервований зміст бітів). Більш докладний опис регістрів наводиться нижче.

Таблиця 7 – Керуючі регістри таймерів/лічильників 0...3

Біти	Опис	Доступ	Значення скидання	Регістр TIMERn/ ім'я та адреса
IR	Регістр переривань. IR може бути записаний, щоб очистити переривання. IR може бути прочитаний, щоб визначити, які з восьми можливих джерел переривань знаходяться в очікуванні.	R/W	0	T0IR – 0xE000 4000 T1IR – 0xE000 8000 T2IR – 0xE007 0000 T3IR – 0xE007 4000

Продовження таблиці 7

Біти	Опис	Дос туп	Зна- чення ски- дання	Регістр TIMERn/ ім'я та адреса
TCR	Регістр керування таймером. TCR використовується для керування функціями таймера/лічильника. Таймер/лічильник може бути відключений або скинутий через TCR.	R/W	0	T0TCR – 0xE000 4004 T1TCR – 0xE000 8004 T2TCR – 0xE007 0004 T3TCR – 0xE007 4004
TC	Таймер/лічильник. 32-х бітний TC збільшується кожні $R + 1$ циклів P_{CLK} . TC керуються через TCR.	R/W	0	T0TC – 0xE000 4008 T1TC – 0xE000 8008 T2TC – 0xE007 0008 T3TC – 0xE007 4008
PR	Регістр масштабування. Якщо лічильник масштабування (нижче) дорівнює цьому значенню, то в наступному такті інкрементується TC та очищається PC.	R/W	0	T0PR – 0xE000 400C T1PR – 0xE000 800C T2PR – 0xE007 000C T3PR – 0xE007 400C
PC	Лічильник масштабування. 32-х бітний PC являє собою лічильник, який збільшується до значення, що зберігається в PR. Коли досягається значення в PR, TC збільшується, а PC буде очищений.	R/W	0	T0PC – 0xE000 4010 T1PC – 0xE000 8010 T2PC – 0xE007 0010 T3PC – 0xE007 4010
MCR	Регістр керування збігом. MCR використовується для генерації переривання, скидання PC, зупинки PC та TC при виявленні збігу.	R/W	0	T0MCR – 0xE000 4014 T1MCR – 0xE000 8014 T2MCR – 0xE007 0014 T3MCR – 0xE007 4014
MR0	Регістр збігу 0. Значення регістра збігу безперервно порівнюється зі значенням таймера/лічильника. Коли два значення рівні, то це через MCR може скидати TC, зупиняти TC і PC, і/або генерувати переривання.	R/W	0	T0MR0 – 0xE000 4018 T1MR0 – 0xE000 8018 T2MR0 – 0xE007 0018 T3MR0 – 0xE007 4018

Продовження таблиці 7

Біти	Опис	Дос- туп	Зна- чення ски- дання	Регістр TIMERn/ ім'я та адреса
MR1	Регістр збігу 1. Дивись опис MR0.	R/W	0	T0MR1–0xE000401C T1MR1–0xE000801C T2MR1–0xE007001C T3MR1–0xE007401C
MR2	Регістр збігу 2. Дивись опис MR0.	R/W	0	T0MR2–0xE0004020 T1MR2–0xE0008020 T2MR2–0xE0070020 T3MR2–0xE0074020
MR3	Регістр збігу 3. Дивись опис MR0.	R/W	0	T0MR3 – 0xE000 4024 T1MR3 – 0xE000 8024 T2MR3 – 0xE007 0024 T3MR3 – 0xE007 4024
CCR	Регістр керування захопленням. CCR програмує сигнали на входах захоплення, які використовуються для завантаження регістрів захоплення, а також генерацію переривання, коли подія захоплення відбувається.	R/W	0	T0CCR – 0xE000 4028 T1CCR – 0xE000 8028 T2CCR – 0xE007 0028 T3CCR – 0xE007 4028
CR0	Регістр захоплення 0. CR0 завантажується значенням TC, коли відбувається подія захоплення на вході CAPn.0(CAP0.0, CAP1.0, CAP2.0 або CAP3.0).	RO	0	T0CR0 – 0xE000 402C T1CR0 – 0xE000 802C T2CR0 – 0xE007 002C T3CR0 – 0xE007 402C

Продовження таблиці 7

Біти	Опис	Дос- туп	Зна- чення ски- дання	Регістр TIMERN/ ім'я та адреса
CR1	Регістр захоплення 1. CR1 завантажується значенням TC, коли відбувається подія захоплення на вході CAPn.0(CAP1.0, CAP1.1, CAP2.1 або CAP3.1).	RO	0	T0CR1 – 0xE000 4030 T1CR1 – 0xE000 8030 T2CR1 – 0xE007 0030 T3CR1 – 0xE007 4030
EMR	Зовнішній регістр збігу EMR керує сигналами на зовнішніх виводах MATn.0...3 при збігу (n відповідає таймерам від 0 до 3).	R/W	0	T0EMR – 0xE000 403C T1EMR – 0xE000 803C T2EMR – 0xE007 003C T3EMR – 0xE007 403C
CTCR	Регістр керування підрахунком CTCR вибирає між режимом таймера і режимом лічильника зовнішніх подій, та в режимі лічильника вибирає активний рівень сигналу для підрахунку.	R/W	0	T0CTCR – 0xE000 4070 T1CTCR – 0xE000 8070 T2CTCR – 0xE007 0070 T3CTCR – 0xE007 4070

3.6 Регістри переривань IR

Регістри переривань мають адреси, які наведено у таблиці 8, та складаються з прапорців переривань при настанні подій збігу та захоплення. Якщо переривання генерується, то відповідний біт в IR буде мати одиничне значення. В іншому випадку, буде мати нульове значення. Записом логічної до відповідного IR-біта останній буде скинутий.

Таблиця 8 – Опис бітів регістрів переривань

Біт	Символ	Опис	Значення скидання
0	MR0 переривання	Прапорець переривання збігу для каналу 0	0
1	MR1 переривання	Прапорець переривання збігу для каналу 1	0
2	MR2 переривання	Прапорець переривання збігу для каналу 2	0
3	MR3 переривання	Прапорець переривання збігу для каналу 3	0
4	CR0 переривання	Прапорець переривання події захоплення для каналу 0	0
5	CR1 переривання	Прапорець переривання події захоплення для каналу 1	0
6	—	Зарезервований	0
7	—	Зарезервований	0

3.7 Регістри керування таймерами T[0/1/2/3]CR

Регістри керування таймерами T[0/1/2/3]CR мають адреси: 0xE000 4004, 0xE000 8004, 0xE007 0004, 0xE007 4004) та використовується для керування роботою таймерів/лічильників.

Нижче наведено опис бітів регістрів керування таймерами T[0/1/2/3]CR (таблиця 9).

Таблиця 9 – Опис бітів регістрів керування таймерами

Біт	Символ	Опис	Значення скидання
0	Дозвіл лічильника	Коли одиниця, таймер/лічильник і лічильник масштабування включені для підрахунку. При нулі, лічильники відключені.	0

Продовження таблиці 9

Біт	Символ	Опис	Значення скидання
1	Скидання лічильника	Коли одиниця, таймер/лічильник і лічильник масштабування включені для підрахунку. При нулі, лічильники відключені.	0
7:2	—	Зарезервовані. Користувачі про грами не повинні записувати до зарезервованих бітів. Значення зчитане із зарезервованого біта не визначене.	NA

3.8 Регістри керування підрахунком

Регістри керування підрахунком T[0/1/2/3]CTCR мають адреси: 0xE000 4070, 0xE000 8070, 0xE007 0070, 0xE007 4070) та вибирають між режимом таймера і режимом лічильника зовнішніх подій, та в режимі лічильника вибирають активний рівень сигналу для підрахунку .

Нижче наведено опис бітів регістрів керування підрахунком CTCR (таблиця 10).

3.9 Регістри таймерів/лічильників T0TC ... T3TC

32-розрядні регістри таймерів/лічильників T0TC ... T3TC мають адреси: 0xE000 4008, 0xE000 8008, 0xE007 0008, 0xE007 4008 та збільшуються, коли лічильник переддільника досягає кінцевого значення. Якщо немає скидання до досягнення свого верхньої межі, таймер/лічильник буде підраховувати через значення 0xFFFF FFFF, а потім повернеться назад до значення 0x0000 0000. Ця подія не викликає переривання, але регістр збігу може бути використаний для виявлення переповнення при необхідності.

Таблиця 10 – Опис бітів регістрів керування підрахунком T[0/1/2/3]CTCR

Біт	Символ	Значення	Опис	Значення скидання
1:0	Режим лічильника/таймера		Це поле вибирає який фронт P_{CLK} може збільшувати лічильник масштабування (РС), або коли очищається РС та інкрементується таймер/лічильник (ТС). Режим таймера: ТС збільшується, коли лічильник масштабування збігається з регістром масштабування.	00
		00	Режим таймера: кожен наростаючий фронт P_{CLK}	
		01	Режим лічильника: ТС збільшується за наростаючим фронтом сигналу на вході CAP, який обрано бітами 3:2.	
		10	Режим лічильника: ТС збільшується за спадаючим фронтом сигналу на вході CAP, який обрано бітами 3:2.	
		11	Режим лічильника: ТС збільшується за обома фронтами сигналу на вході CAP, який обрано бітами 3:2.	
3:2	Вибір входу для підрахунку		Коли біти 1:0 в цьому регістрі не дорівнюють 00, ці біти визначають, який вивід CAP обирається для підрахунку:	00
		00	CAPn.0 для TIMERN	
		01	CAPn.1 для TIMERN Примітка: Якщо режим лічильника обраний для конкретного входу CAPn в TnCTCR, 3 біти для цього входу в регістрі керування захопленням (TnCCR) повинні бути за прогамовані як 000. Однак, захоплення і/або переривання можуть бути обрані для іншого входу CAPn в цьому таймері	
		10	Зарезервований	
		11	Зарезервований	
7:4	—	—	Зарезервовані. Користувачські програми не повинні записувати до зарезервованих бітів. Значення зчитане із зарезервованого біта не визначене	NA

3.10 Регістри масштабування T0PR ... T3PR

32-розрядні регістри масштабування T0PR ... T3PR мають адреси: , 0xE000 400C, 0xE000 800C, 0xE007 000C, 0xE007 400C та визначають максимальне значення для лічильника масштабування.

3.11 Регістри лічильників масштабування T0PC ... T3PC

32-розрядні лічильники масштабування T0PC...T3PC мають адреси: 0xE000 4010, 0xE000 8010, 0xE007 0010, 0xE007 4010 та керують зміною коефіцієнта ділення частоти P_{CLK} , перш ніж вона застосовується до таймера/лічильника. Це дозволяє керувати роздільною здатністю таймера в порівнянні з максимальним часом до переповнення таймера. Лічильник масштабування збільшується при кожному такті P_{CLK} . Коли він досягає значення, що зберігається в регістрі масштабування, таймер/лічильник збільшується на одиницю і лічильник масштабування скидається при наступному такті P_{CLK} . Це призводить до того, що таймер/лічильник інкрементується за кожним імпульсом P_{CLK} при $PR = 0$, інкрементується кожні 2 такти P_{CLK} при $PR = 1$ і т.д.

3.12 Регістри збігу MR0 ... MR3

Регістри збігу мають адреси, які наведено в таблиці 7. Значення відповідного регістра збігу безперервно порівнюється зі значенням таймера/лічильника. Коли ці два значення рівні, то відбуваються дії, які керуються значеннями в регістрі MCR. Можливі дії: генерація переривання, скидання таймера/лічильника або зупинки таймера.

3.13 Регістри керування збігом T[0/1/2/3]MCR

Регістри керування збігом мають адреси: 0xE000 4014, 0xE000 8014, 0xE007 0014, 0xE007 4014 та використовується для керування відповідною операцією, коли значення одного з регістрів збігу відповідає значенню таймера/лічильника. Опис кожного з бітів регістрів наведено в таблиці 11.

Таблиця 11 – Опис бітів регістрів керування збігом T[0/1/2/3]MCR

Біт	Символ	Значення	Опис	Значення скидання
0	MR0I	1	Переривання за MR0: переривання генерується, коли MR0 збігається зі значенням в ТС.	0
		0	Це переривання заборонено.	
1	MR0R	1	Скидання за MR0: ТС буде скинуто, якщо MR0 збігається з ним.	0
		0	Функція відключена.	
2	MR0S	1	Зупинка за MR0: ТС і РС будуть зупинені і TCR[0] буде скинуто в 0, якщо MR0 збігається з ТС.	0
		0	Функція відключена.	
3	MR1I	1	Переривання за MR1: переривання генерується, коли MR1 збігається зі значенням в ТС.	0
		0	Це переривання заборонено	
4	MR1R	1	Скидання за MR1: ТС буде скинуто, якщо MR1 збігається з ним.	0
		0	Функція відключена.	
5	MR1S	1	Зупинка за MR1: ТС і РС будуть зупинені і TCR[0] буде скинуто в 0, якщо MR1 збігається з ТС.	0
		0	Функція відключена.	

Продовження таблиці 11

Біт	Символ	Значення	Опис	Значення скидання
6	MR2I	1	Переривання за MR2: переривання генерується, коли MR2 збігається зі значенням в ТС.	0
		0	Це переривання заборонено.	
7	MR2R	1	Скидання за MR2: ТС буде скинуто, якщо MR2 збігається з ним.	0
		0	Функція відключена.	
8	MR2S	1	Зупинка за MR2: ТС і РС будуть зупинені і TCR[0] буде скинуто в 0, якщо MR2 збігається з ТС.	0
		0	Функція відключена.	
9	MR3I	1	Переривання за MR3: переривання генерується, коли MR3 збігається зі значенням в ТС.	0
		0	Це переривання заборонено	
10	MR3R	1	Скидання за MR3: ТС буде скинуто, якщо MR3 збігається з ним.	0
		0	Функція відключена.	
11	MR3S	1	Зупинка за MR3: ТС і РС будуть зупинені і TCR[0] буде скинуто в 0, якщо MR3 збігається з ТС.	0
		0	Функція відключена.	
15:12	—		Зарезервовані. Користувачські програми не повинні записувати до зарезервованих бітів. Значення зчитане із зарезервованого біта не визначене.	NA

3.14 Регістри захоплення CR0 та CR1

Кожен регістр захоплення CR0 та CR1 має адреси, які наведено в таблиці 7, та пов'язаний із відповідним виводом пристрою і може бути завантажений значенням із таймера/лічильника, коли відбувається певна подія на цьому виводі. Налаштування в регістрі керування захопленням визначає, чи включена функція захоплення, і подія захоплення відбувається за переднім фронтом, заднім фронтом або за обома фронтами сигналу на відповідному виводі.

3.15 Регістри керування захопленням T[0/1/2/3]CCR

Регістри керування захоплення T[0/1/2/3]CCR мають адреси: 0xE000 4028, 0xE000 8028, 0xE007 0028, 0xE007 4028 та використовується для керування одного з двох регістрів захоплення, які завантажуються значенням із таймера/лічильника, коли відбувається подія захоплення і генерується переривання. Встановлення обох бітів для зростаючого та спадаючого фронтів одночасно є дійсною конфігурацією, в результаті чого подія захоплення відбувається для обох фронтів. У наведеному нижче описі, "n" представляє номер таймера, від 0 до 3.

Примітка: Якщо режим лічильника обраний для конкретного входу CAP в CTR, 3 біти для цього входу в цьому регістрі повинні бути запрограмовані як 000, але захоплення і/або переривання можуть бути обрані для іншого входу CAP.

Нижче наведено опис бітів регістра керування захопленням CCR (таблиця 12).

Таблиця 12– Опис бітів регістрів керування захопленням T[0/1/2/3]CCR

Біт	Символ	Значення	Опис	Значення скидання
0	CAP0RE	1	Захоплення CAPn.0 за переднім фронтом: послідовність із 0 в 1 на CAPn.0 змусить CR0 бути завантаженим вмістом TC.	0
		0	Функція відключена.	
1	CAP0FE	1	Захоплення CAPn.0 за заднім фронтом: послідовність із 1 в 0 на CAPn.0 змусить CR0 бути завантаженим вмістом TC.	0
		0	Функція відключена.	
2	CAP0I	1	Переривання CAPn.0 за подією: завантаження CR0 завдяки події CAPn.0 буде генерувати переривання.	0
		0	Функція відключена.	

Продовження таблиці 12

Біт	Символ	Значення	Опис	Значення скидання
3	CAP1RE	1	Захоплення CAPn.1 за переднім фронтом: послідовність із 0 в 1 на CAPn.1 змусить CR1 бути завантаженим вмістом ТС.	0
		0	Функція відключена.	
4	CAP1FE	1	Захоплення CAPn.1 за заднім фронтом: послідовність із 1 в 0 на CAPn.1 змусить CR1 бути завантаженим вмістом ТС.	0
		0	Функція відключена.	
5	CAP1I	1	Переривання CAPn.1 за подією: завантаження CR1 завдяки події CAPn.1 буде генерувати переривання.	0
		0	Функція відключена.	
15:6	—		Зарезервовані. Користувачькі програми не повинні записувати до зарезервованих бітів. Значення зчитане із зарезервованого біта не визначене.	NA

3.16 Регістри зовнішнього збігу T[0/1/2/3]EMR

Регістри зовнішнього збігу T[0/1/2/3]EMR мають адреси: 0xE000 403C, 0xE000 803C, 0xE007 003C, 0xE007 403C та керують сигналами на зовнішніх виводах MATn.0...3 при збігу. У наведеному нижче описі, "n" представляє номер таймера, від 0 до 3, а "m" представляє номер збігу, від 0 до 3.

Нижче наведено опис бітів регістрів зовнішнього збігу EMR (таблиця 13).

Таблиця 13 – Опис бітів регістрів зовнішнього збігу T[0/1/2/3]EMR

Біт	Символ	Опис	Значення скидання
0	EM0	Зовнішній збіг 0. Коли відбувається збіг між ТС і MR0, цей біт відображає зміну сигналу на зовнішньому виводі: на низький рівень, на високий рівень, або нічого не робити, в залежності від бітів 5:4 цього регістра. Сигнал на виводі MATn.0 формується в додатній логіці: 0 = низький рівень, 1 = високий рівень.	0

Продовження таблиці 4.13

Біт	Символ	Опис	Значення скидання
1	ЕМ1	Зовнішній збіг 1. Коли відбувається збіг між ТС і MR0, цей біт відображає зміну сигналу на зовнішньому виводі: на низький рівень, на високий рівень, або нічого не робити, в залежності від бітів 7:6 цього регістра. Сигнал на виводі MATn.0 формується в додатній логіці: 0 = низький рівень, 1 = високий рівень.	0
2	ЕМ2	Зовнішній збіг 2. Коли відбувається збіг між ТС і MR0, цей біт відображає зміну сигналу на зовнішньому виводі: на низький рівень, на високий рівень, або нічого не робити, в залежності від бітів 9:8 цього регістра. Сигнал на виводі MATn.0 формується в додатній логіці: 0 = низький рівень, 1 = високий рівень.	0
3	ЕМ3	Зовнішній збіг 3. Коли відбувається збіг між ТС і MR0, цей біт відображає зміну сигналу на зовнішньому виводі: на низький рівень, на високий рівень, або нічого не робити, в залежності від бітів 11:10 цього регістра. Сигнал на виводі MATn.0 формується в додатній логіці: 0 = низький рівень, 1 = високий рівень.	0
5:4	ЕМC0	Зовнішнє керування збігом 0. Визначає функції зовнішнього збігу 0. Таблиця 4.14 показує кодування цих бітів.	00
7:6	ЕМC1	Зовнішнє керування збігом 1. Визначає функції зовнішнього збігу 1. Таблиця 4.14 показує кодування цих бітів.	00
9:8	ЕМC2	Зовнішнє керування збігом 2. Визначає функції зовнішнього збігу 2. Таблиця 4.14 показує кодування цих бітів.	00
11:10	ЕМC3	Зовнішнє керування збігом 3. Визначає функції зовнішнього збігу 3. Таблиця 4.14 показує кодування цих бітів.	00
15:12	—	Зарезервовані. Користувачькі програми не повинні записувати до зарезервованих бітів. Значення зчитане із зарезервованого біту не визначене.	NA

Нижче наведено функції бітів керування зовнішнього збігу ЕМС (таблиця 14).

Таблиця 14– Функції бітів керування зовнішнім збігом ЕМС

EMR[11:10], EMR[9:8], EMR[7:6] або EMR[5:4]	Функція
00	Нічого не робиться.
01	Очистити відповідний біт зовнішнього збігу/вихід в 0 (MATn.m на виводі низький рівень, якщо під'єднаний).
10	Встановити відповідний біт зовнішнього збігу/вихід в 1 (MATn.m на виводі високий рівень, якщо під'єднаний).
11	Перемикання відповідного зовнішнього збігу/виходу.

3.17 Структура модуля таймерів/лічильників

Структуру модуля таймерів/лічильників наведено на рисунку 8.

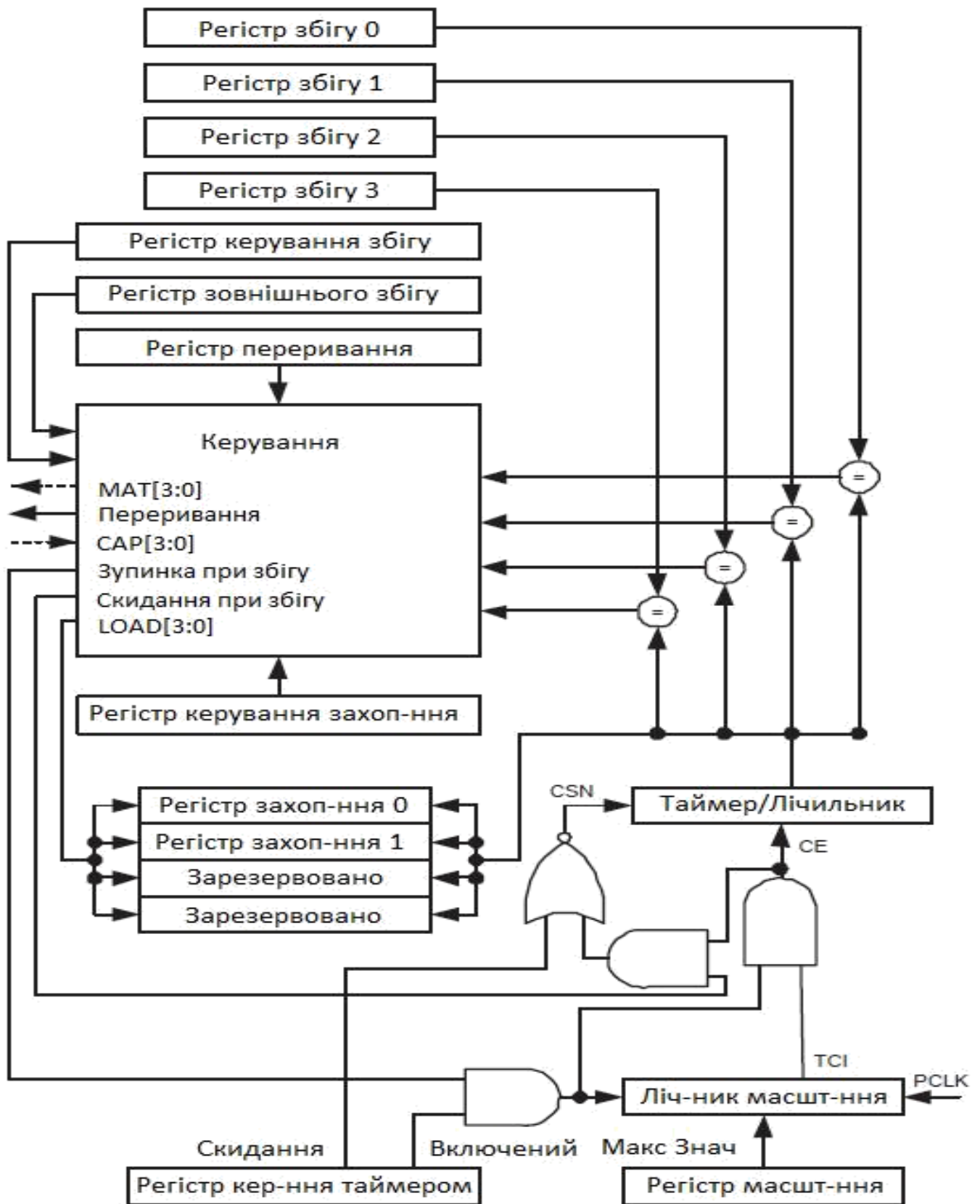


Рисунок 8 – Структурна схема модуля таймерів/лічильників

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) Скільки таймерів є у складі мікроконтролерів сімейства LPC2300?
- 2) Опишіть спрощену структуру таймера загального призначення.
- 3) Від чого залежить швидкість лічби таймера?
- 4) Скільки каналів захоплення має кожен таймер?
- 5) Опишіть роботу таймера в режимі захоплення.
- 6) Опишіть роботу таймера в режимі лічильника зовнішніх подій.
- 7) Опишіть роботу таймера в режимі збігу.
- 8) Скільки каналів збігу має кожен таймер?
- 9) Назвіть та опишіть керуючі регістри модуля таймерів.
- 10) Дайте опис бітів регістрів переривань.
- 11) Дайте опис бітів регістрів керування таймерами.
- 12) Дайте опис бітів регістрів керування підрахунком.
- 13) Дайте опис бітів регістрів таймерів/лічильників.
- 14) Дайте опис бітів регістрів масштабування та лічильників масштабування.
- 15) Дайте опис бітів регістрів збігу та керування збігом.
- 16) Дайте опис бітів регістрів захоплення та керування захопленням.
- 17) Дайте опис бітів регістрів зовнішнього збігу.
- 18) Опишіть структуру модуля таймерів/лічильників.
- 19) За яким фронтом зовнішнього сигналу може відбуватися подія захоплення?
- 20) Як може змінюватися сигнал на відповідному виході при настанні події збігу?
- 21) Які прапорці переривань можуть використовуватись при програмуванні роботи таймерів?

4.1 Загальна характеристика

На перший погляд, модуль ШІМ здається набагато складнішим, ніж таймери загального призначення. Насправді він представляє собою той же таймер, який оснащено додатковими блоками. Модуль ШІМ (рисунок 9) забезпечує формування до шести ШІМ – сигналів з керуванням за одним фронтом чи до трьох сигналів – з керуванням за обома фронтами.

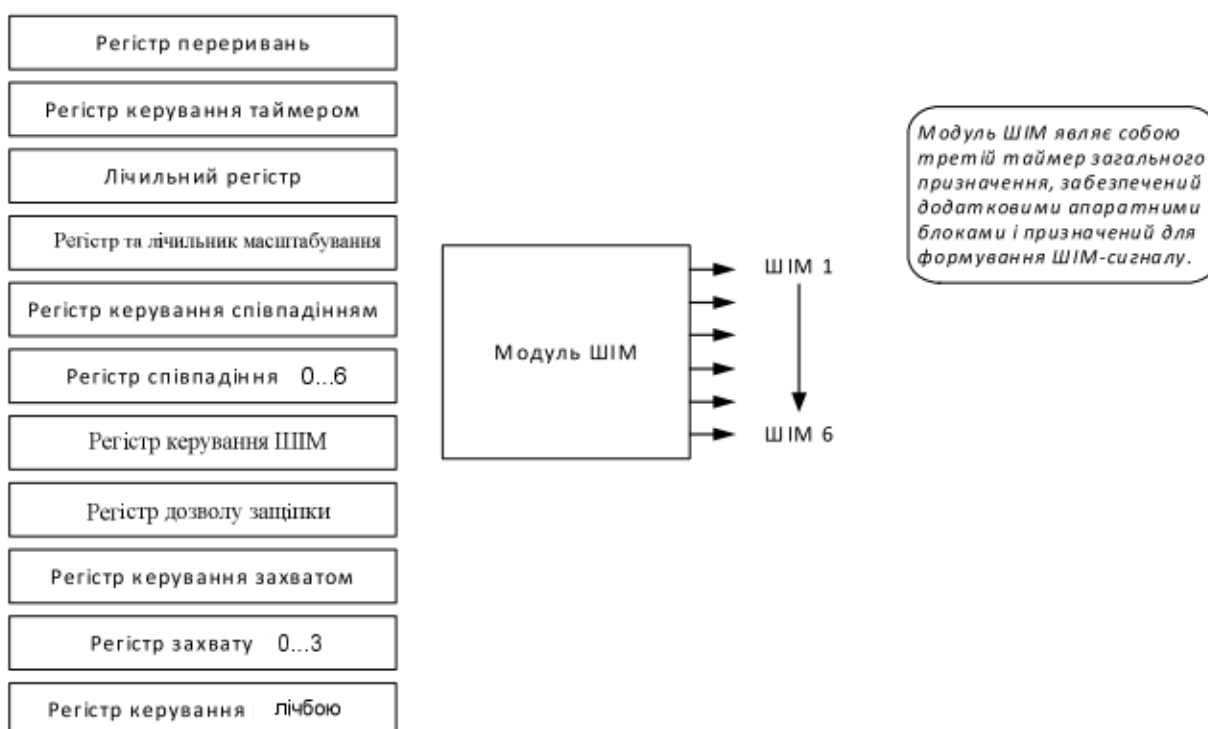


Рисунок 9 – Модуль ШІМ

У таймерах загального призначення при записі нового числа в регістр збігу це число починає діяти відразу ж. Якщо не передбачити спеціальних заходів в програмі, це може викликати спотворення ШІМ – сигналу. При одночасному оновленні декількох каналів нові значення будуть вступати в дію в різні моменти формованого імпульсу, що може привести до непередбачуваних результатів. Щоб уникнути подібних ситуацій, в модулі ШІМ передбачений спеціальний механізм заціпок (рисунок 10), який дозволяє змінювати значення ШІМ «на льоту», при цьому нові значення будуть вступати в силу одночасно і тільки на початку нового періоду.

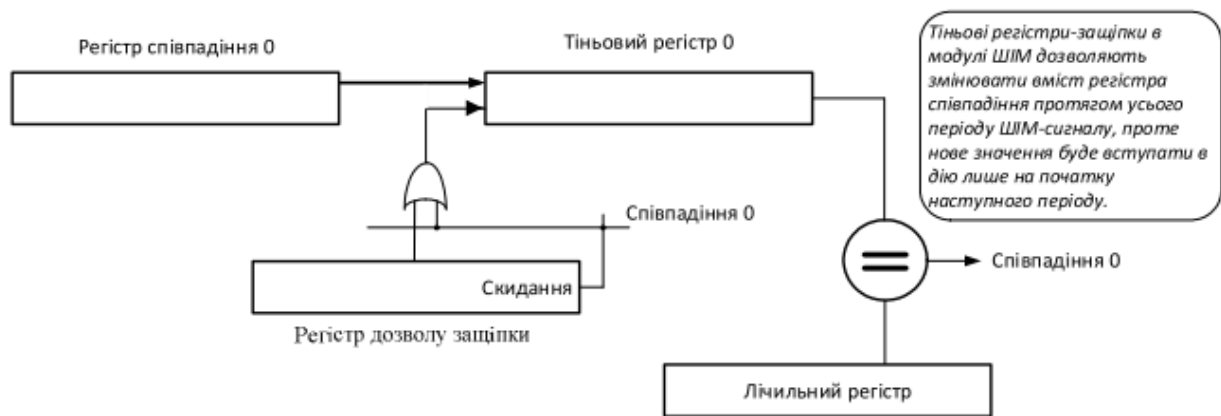


Рисунок 10 – Регістр – защипка

Вміст будь-якого з реєстрів збігу може бути змінено в будь-який момент часу, однак він не буде використовуватися до тих пір, поки не буде встановлений відповідний цьому каналу біт в реєстрі дозволу защипки (Latch Enable Register – LER).

Після встановлення зазначеного біта вміст реєстра співпадіння буде скопійовано в тіньовий реєстр на початку наступного періоду. Таким чином, гарантується, що всі зміни реєстрів будуть здійснюватися одночасно на початку періоду. Крім цих тіньових реєстрів-защипок, канали збігу модуля ШІМ функціонують точно так само, як і в таймерах.

Другою відмінністю модуля ШІМ від звичайних таймерів є схема керування виводами мікроконтролера. Канали співпадіння здійснюють керування виводами не безпосередньо, а через RS– тригери (рисунок 11).

Така схема з RS– тригерів і мультиплексорів дозволяє формувати сигнали, які керуються за одним чи за обома фронтами. Стан мультиплексорів визначається реєстром керування PWMPCR. За допомогою цих мультиплексорів можна задавати одну з двох конфігурацій вихідних каскадів модуля.

Перша конфігурація відповідає модуляції з керуванням за одним фронтом (рисунок 12).

У цій конфігурації вихід 0– го каналу збігу підключається до входу S всіх тригерів, а виходи інших каналів – до входів R відповідних тригерів. Таким чином, 0– й канал визначає тривалість періоду ШІМ– сигналів, загальну для всіх

каналів. В кінці періоду цей канал скидає лічильний регістр модуля і встановлює свій вихід в 1.

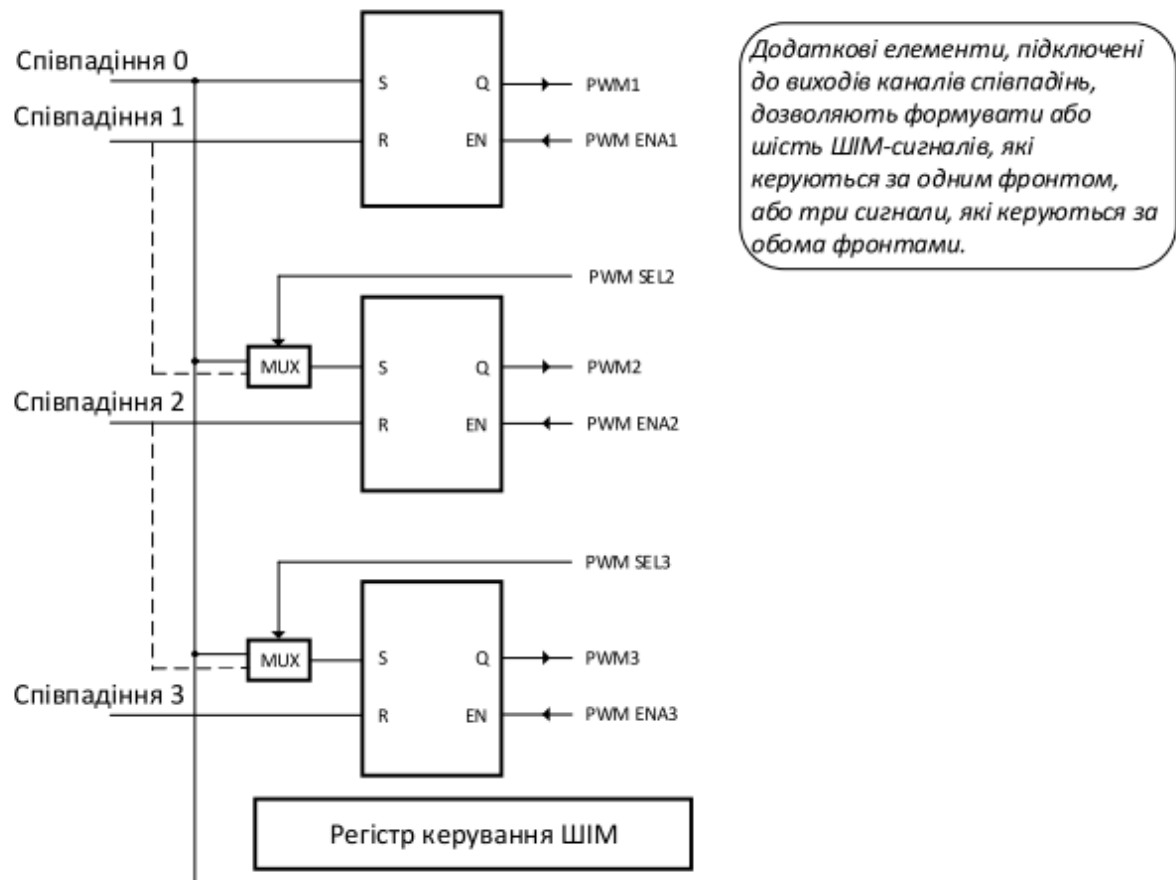


Рисунок 11 – Керування виводами в модулі ШІМ

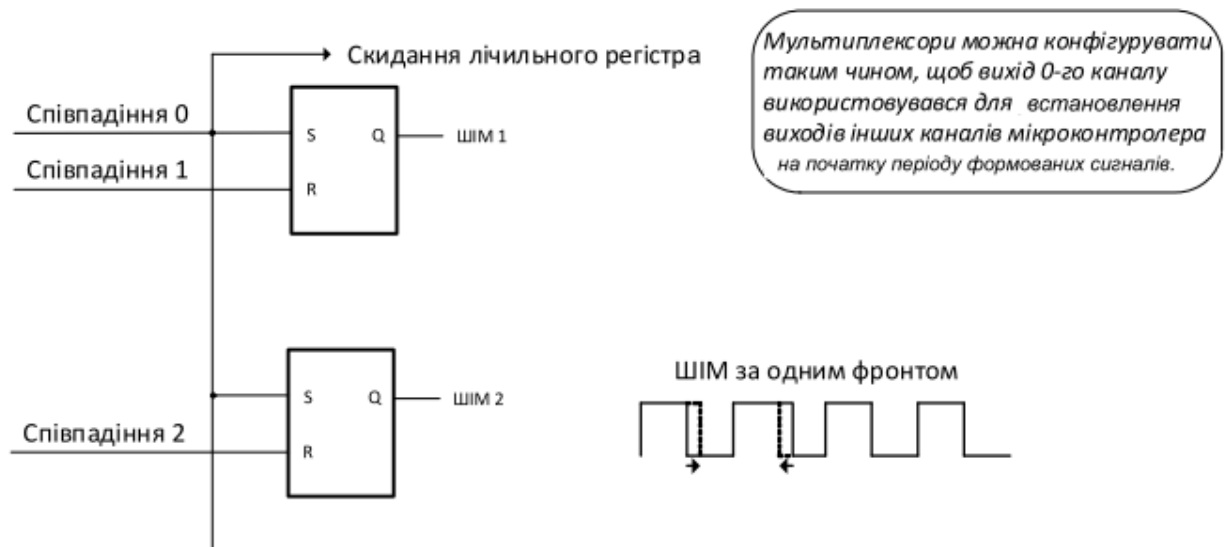


Рисунок 12 – Модуляція за одним фронтом

В результаті всі тригери встановлюються на початку періоду ШІМ-сигналу. При цьому на їх виходах Q з'являється напруга ВИСОКОГО рівня, а на виводах мікроконтролера формується наростаючий фронт. Модуляція сигналу здійснюється іншими каналами співпадіння, при цьому кожен канал співпадіння відповідає одному каналу ШІМ. При збігу одного з каналів відповідний тригер скидається і на виході каналу ШІМ формується спадаючий фронт. Таким чином, модуляція здійснюється зміною вмісту регістра необхідного каналу збігу.

Друга конфігурація мультиплексорів відповідає модуляції ШІМ – сигналів з керуванням за обома фронтами (рисунк 13). У цій конфігурації 0 – й канал збігу використовується виключно для скидання лічильного регістра модуля в кінці кожного періоду ШІМ- сигналу, а входи R і S тригерів підключені до виходів відповідних каналів збігу. На початку періоду на виході каналу ШІМ присутня напруга НИЗЬКОГО рівня. Наростаючий фронт імпульсу формується каналом збігу, підключеним до входу S тригеру, а спадаючий фронт – каналом, підключеним до входу R.

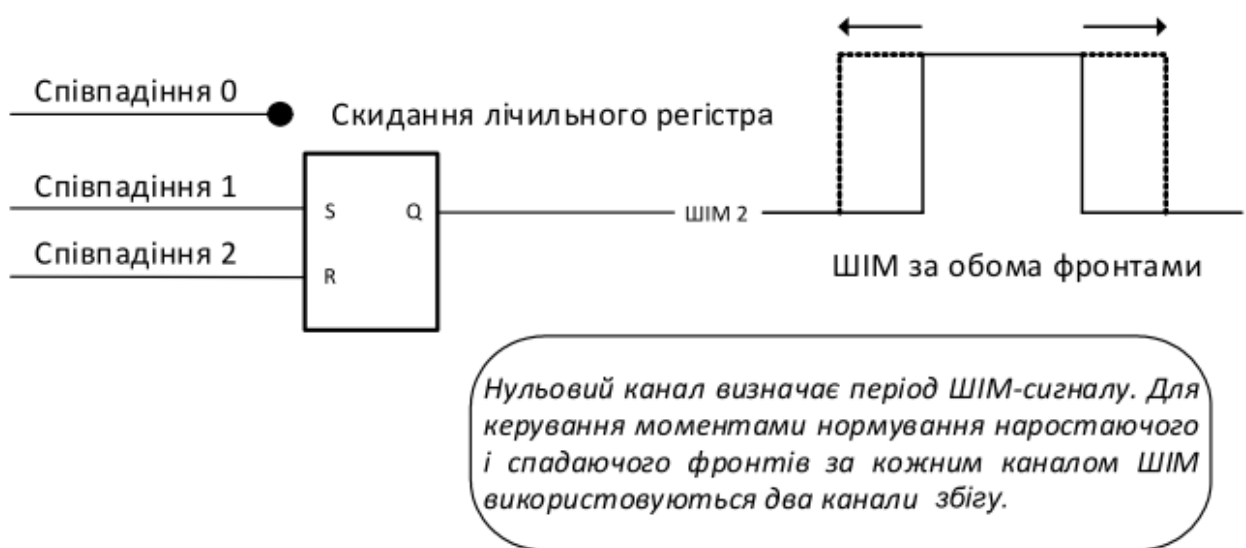


Рисунок 13 – Модуляція за обома фронтами

Зверніть увагу: регістр PWMPCR використовується для того, щоб гарантувати встановлення виходів каналів збігу при настанні відповідних подій. Якщо в цьому регістрі буде записано некоректне значення, модуль ШІМ працювати не буде. Крім того, на відміну від базового таймера, для роботи модуля ШІМ не потрібно ніяких переривань.

4.2 Режим лічильника

Як і таймери загального призначення, модуль ШІМ теж може працювати в режимі лічильника зовнішніх подій. Для керування режимом використовується регістр керування лічильником. За допомогою цього регістра можна змінити тактовий сигнал модуля з Pclk на сигнал від зовнішнього джерела, що подається на вхід захоплення модуля. При цьому лічильний регістр може інкрементуватися за наростаючим, спадаючим або навіть за обома фронтами зовнішнього сигналу.

4.3 Структурна схема модуля широтно–імпульсного модулятора

На рисунку 14 наведено структурну схему модуля ШІМ. Частини, які були додані до стандартного блоку таймера (рисунок 12), наведено на правій стороні і у верхній частині рисунка 14.

Модуль ШІМ (Pulse width modulator–PWM) заснований на стандартному блоці таймера і успадковує всі його функції, хоча окрема функція ШІМ закріплена на LPC2300. Таймер призначений для підрахунку періодів тактової частоти периферії (P_{CLK}) і при необхідності повинен генерувати переривання або виконувати інші дії при настанні певних значень таймера на підставі семи регістрів збігу. Функція ШІМ додається до цих функцій і заснована на регістрах подій збігу. Здатність окремо керувати зростанням і спаданням фронтів вихідних імпульсів дозволяє PWM використовуватися для значної кількості застосувань. Наприклад, для керування багатофазним двигуном, як правило, потрібно три ШІМ, які не перекривають один одного, з індивідуальним керуванням ширини імпульсів і позицій всіх трьох каналів. Два регістри збігу можуть бути використані для забезпечення одного фронту керування виходу ШІМ. Один регістр збігу (PWMMR0) керує шпаруватістю сигналу ШІМ, шляхом скидання лічильника при збігу. Інший регістр збігу керує положенням фронту PWM. Додаткові сигнали ШІМ вимагають тільки один зареєстрований збіг, так як частота повторення однакова для всіх виходів ШІМ. Декілька поодиноких фронтів керуються ШІМ – виходами, які повинні мати наростаючий фронт на початку кожного циклу ШІМ, коли відбувається збіг PWMMR0. Три регістри збігу можуть бути використані для забезпечення виведення ШІМ за обома фронтами. Знову ж, регістр збігу PWMMR0 контролює частоту ШІМ. Інші

реєстри збігу керують двома сигналами PWM. Спеціальні реєстри забезпечують керування за подвійними фронтами, як за переднім так і за заднім фронтом. Це дозволяє працювати як із додатними так і з від'ємними імпульсами.

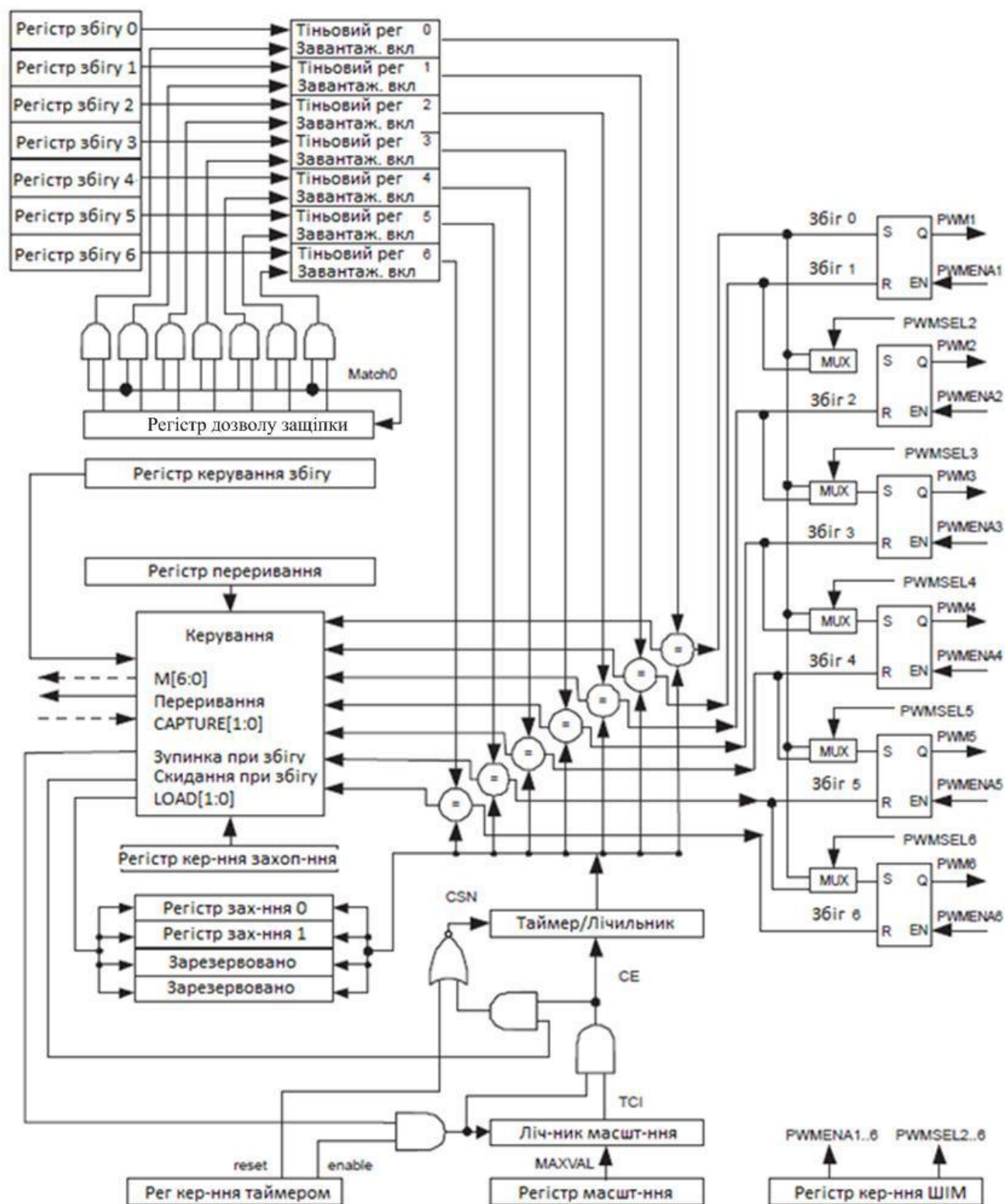


Рисунок 14 – Структурна схема модуля ШІМ

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) Дайте визначення широтно–імпульсній модуляції.
- 2) Як розраховується шпаруватість ШІМ–сигналу?
- 3) Які ШІМ–сигнали може формувати модуль ШІМ?
- 4) Який таймер загального призначення виконує функцію модуля ШІМ?
- 5) Який механізм введено в модуль ШІМ з метою недопущення спотворення ШІМ–сигналу?
- 6) В які моменти періоду формування ШІМ–сигналу вступають в силу нові значення при оновленні відповідного каналу?
- 7) Який регістр та як використовується при реалізації механізму недопущення спотворення ШІМ–сигналу?
- 8) Які відмінності існують у функціонуванні каналів збігу модуля ШІМ крім наявності тіньових регістрів–заціпок?
- 9) Чим відрізняється схема керування виводами модуля ШІМ від звичайних таймерів?
- 10) Поясніть роботу схеми керування виводами в модулі ШІМ.
- 11) Поясніть особливості широтно–імпульсної модуляції за одним фронтом.
- 12) Поясніть особливості широтно–імпульсної модуляції за обома фронтами.
- 13) Які регістри використовуються для керуванням модулем ШІМ?
- 14) Поясніть використання модуля ШІМ в режимі лічильника.
- 15) Яка тактова частота використовується модулем ШІМ?
- 16) Скільки регістрів збігу має модуль ШІМ?
- 17) Як можуть змінюватись фронти вихідних імпульсів в модулі ШІМ при настанні події збігу?
- 18) Який регістр використовується для керування шпаруватістю?
- 19) Який регістр використовується для керування положенням фронту PWM?
- 20) Який регістр використовується для керування частотою ШІМ–сигналу?

5 Сторожовий таймер

5.1 Загальні відомості

Як і більшість інших мікроконтролерів, мікроконтролери сімейства LPC2300 мають в своєму складі сторожовий таймер, який використовується для встановлення контролю над програмою в разі її фатального збою. Як тактовий сигнал сторожового таймера може використовуватися сигнал від вбудованого RC-генератора, сигнал від генератора модуля RTC або системний тактовий сигнал Pclk.

4.5.2 Період сторожового таймера

Сторожовий таймер має чотири керуючих регістри. Період тайм-ауту сторожового таймера визначається вмістом регістра постійної таймера WDTC. Значення періоду тайм-ауту визначається за формулою:

$$Wdperiod = Pclk \times WDTC \times 4.$$

Вбудований сторожовий таймер може викликати скидання процесора або згенерувати переривання. При скиданні за сторожовим таймером встановлюється спеціальний прапорець, так що програма може блокувати виконання «м'якого» скидання.

Мінімальне значення регістра WDTC дорівнює 256, а максимальне – 232. Таким чином, при тактовій частоті процесора 60 МГц мінімальний період сторожового таймера дорівнює 17, 066 мкс, а максимальний – трохи менше за 5 хвилин.

Після запису значення постійної сторожового таймера можна приступати до конфігурації його режиму роботи. Регістр режиму сторожового таймера WDMOD (рисунок 15) містить три керуючих біти, два з яких визначають, чи буде сторожовий таймер генерувати переривання і сигнал скидання процесора, а третій використовується для дозволу роботи сторожового таймера.

Крім того, в регістрі керування є два прапорці. Один з них, WDTOF, встановлюється при тайм-ауті сторожового таймера і скидається тільки після зовнішнього апаратного скидання.



Рисунок 15 – Регістр режиму сторожового таймера

Це дозволяє в стартовому коді програми визначити, що стало причиною скидання – включення живлення або помилка в програмі. Другий прапорець є прапорцем переривання сторожового таймера. Даний прапорець доступний тільки для читання, а для скидання його необхідно прочитати. Якщо ви збираєтеся налагоджувати програму при включеному сторожовому таймері, не дозволяйте скидання від нього, оскільки це призведе до відключення JTAG – відлагоджувача при тайм-ауті сторожового таймера.

Після завдання періоду тайм-ауту і режиму роботи сторожового таймера його можна включити, записавши в регістр запуску WDFEED певну послідовність чисел, аналогічну послідовності, яка необхідна для включення схеми ФАПЧ. Для запуску сторожового таймера треба спочатку записати число 0AA, а потім число 0x55. Якщо ця послідовність буде порушена, відбудеться помилка запуску сторожового таймера, яка викличе тайм-аут сторожового таймера, що супроводжується відповідною генерацією переривання/скидання. Також необхідно відзначити, що хоча сторожовий таймер може бути дозволений у регістрі режиму, він не почне працювати до першого запису необхідної послідовності в регістр запуску. Щоб уникнути спрацьовування сторожового таймера після того, як він буде запуснений, його необхідно періодично перевстановити, записуючи ту ж послідовність чисел.

Останнім регістром сторожового таймера є регістр значення WDTV, який дозволяє дізнатися поточне значення сторожового таймера.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) Поясніть призначення сторожового таймера.
- 2) Який сигнал може використовуватися в якості тактового для сторожового таймера?
- 3) Скільки керуючих регістрів має сторожовий таймер?
- 4) Як обчислюється період тайм-ауту сторожового таймера?
- 5) Які дії може викликати сторожовий таймер після закінчення тайм-ауту?
- 6) Чому дорівнює мінімальний та максимальний період сторожового таймера при тактовій частоті процесора 60 МГц?
- 7) Опишіть призначення регістра постійної таймера WDTC.
- 8) Опишіть призначення регістра режиму сторожового таймера WDMOD.
- 9) Як в стартовому коді програми визначити, що стало причиною скидання – включення живлення або помилка в програмі?
- 10) Як можна скинути прапорець переривання сторожового таймера?
- 11) Як можна запобігти відключенню JTAG-відлагоджувача при тайм-ауті сторожового таймера?
- 12) Який регістр використовується для включення сторожового таймера?
- 13) Які дії треба виконати при програмуванні запуску сторожового таймера?
- 14) Що станеться з роботою сторожового таймера в разі порушення відповідної послідовності запису у регістр WDFEED?
- 15) Коли сторожовий таймер починає працювати після його дозволу у регістрі режиму?
- 16) Що треба зробити для уникнення спрацьовування сторожового таймера після того, як він буде запущений?
- 17) Опишіть призначення регістра WDTV.