

## Лекція 8

### Система переривань

#### 1 Загальні відомості

Поняття переривання (від англ. Interrupt) з'явилося в мікропроцесорній техніці давно і з плином часу міцно в ній закріпилося. Перериванням в мікроконтролерах відведена особлива роль. Вони дозволяють мікроконтролеру швидко реагувати на певні події (від англ. Event), підвищуючи тим самим його оперативність. Переривання виникає в результаті певної очікуваної події і тимчасово припиняє виконання основного коду програми, переходячи до виконання підпрограми, яка знаходиться в обробнику переривання. Після завершення цієї підпрограми триває виконання основного коду програми.

Існує поняття зовнішніх і внутрішніх переривань. Зовнішнє переривання ініціюється зовнішньою подією, наприклад натисненням кнопки користувачем або замиканням контактів датчика. Внутрішнє переривання обумовлено деякою подією, пов'язаною, наприклад, з прийомом або передачею даних через інтерфейс, завершенням рахунку таймера і тому подібним.

Розрізняють апаратне і програмне переривання. Апаратне переривання виникає в результаті впливу зовнішніх сигналів або апаратного встановлення регістрових прапорців подій. Програмне переривання формується за допомогою коду програми шляхом програмного встановлення прапорців подій або безпосереднього виклику підпрограми обробки переривання.

Нижче ми познайомимося з двома джерелами виняткових ситуацій: переривання загального призначення (IRQ) і швидке переривання (FIQ) мікроконтролерів ARM7 [1, 4, 7]. Ці винятки використовуються для підтримки різних джерел переривань, зовнішніх по відношенню до ЦПП ARM7. У нашому випадку такими джерелами є користувальницькі периферійні пристрої. Для вивчення структури переривань сімейства LPC2300 нам буде потрібно просте джерело переривань.

В якості такого джерела можна скористатися виводами зовнішніх переривань. Цей периферійний модуль найпростіше конфігурувати. Крім того, до входу EINT1 може бути підключена кнопка, за допомогою якої ми зможемо ініціювати переривання і спостерігати реакцію у налагоджувачі.

## 2. Блок керування виводами

Всі контакти введення/виведення в мікроконтролерах сімейства LPC2300 мають додаткові функції і підключені до вбудованих периферійних модулів через мультиплексори, які називають блоками вибору функції виводів (pinselectblock). За допомогою цього блоку (рисунок 3.19) можна задіяти вивід в якості лінії порту введення/виведення загального призначення (GPIO) або ж «повісити» на нього одну з трьох функцій.

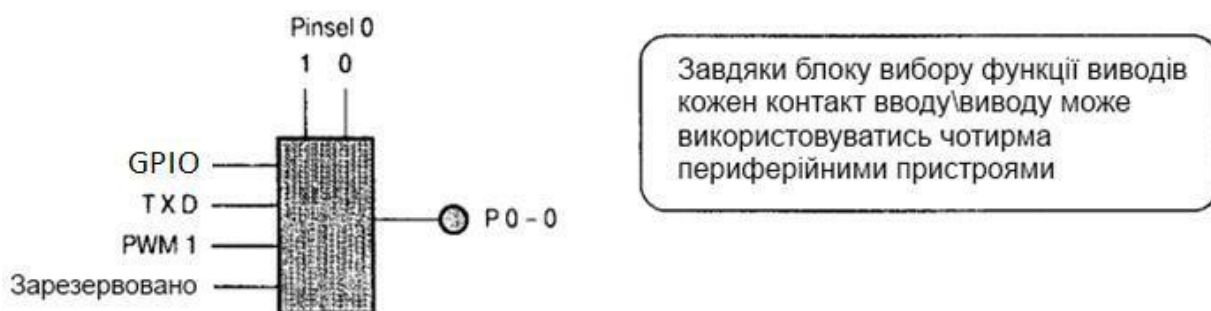


Рисунок 3.19 – Блок вибору функції виводів

При скиданні всі контакти введення/виведення задіяні в якості ліній портів введення/виведення загального призначення. Додаткові функції вибираються за допомогою регістрів PINSEL. Наприклад, лінія зовнішнього переривання EINT0 є одночасно контактом порту введення/виведення загального призначення P2.10. Утримування цієї лінії в стані НИЗЬКОГО рівня при скиданні мікроконтролера викликає запуск вбудованого обробника команд завантажувача. Тому для використання зовнішнього переривання ми повинні записати в регістр вибору функції виведення відповідне значення, що підключає вивід до схеми переривань.

## 3. Виводи зовнішніх переривань

Керування зовнішніми перериваннями здійснюється за допомогою чотирьох регістрів. Регістром EXMODE задається тип переривання – за фронтом або за рівнем. Якщо використовується переривання за фронтом, то в регістрі EXPOL задається активний фронт (наростаючий або спадаючий), при виявленні якого буде генеруватися переривання. Якщо використовується переривання за рівнем, то воно може виникнути лише при НИЗЬКОМУ у рівні на вході

переривання. За допомогою регістра EXWAKE можна дозволити вихід процесора з режиму Power Down за зовнішнім перериванням (рисунок 3.20). Для ініціалізації навчальної програми потрібно задати для переривання EINT0 чутливість за рівнем і підключити цей вивід до процесора за допомогою регістра PINSEL0.

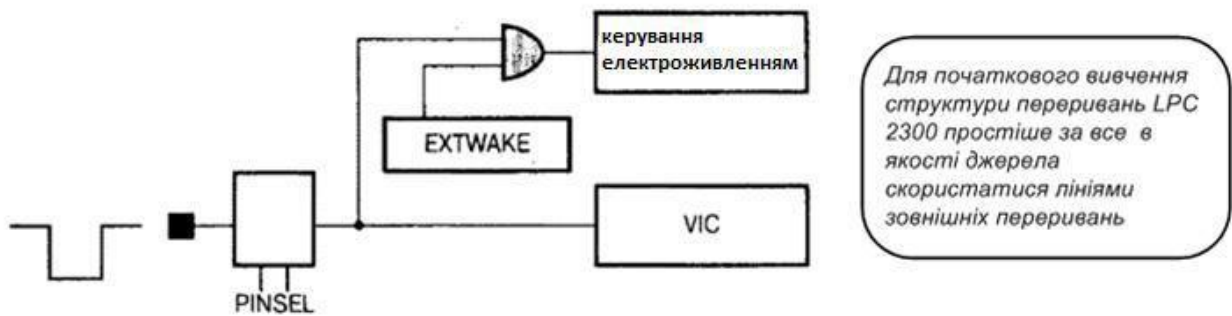


Рисунок 3.20 – Конфігурування лінії зовнішнього переривання

#### 4. Структура переривань

ЦПП ARM7 має дві лінії зовнішніх переривань: одну для режиму оброблення запиту на швидке переривання (FIQ), а іншу для режиму обробки запиту на звичайне переривання (IRQ). В системах на базі ARM7 повинно бути тільки одне джерело переривань, що генерує переривання FIQ, щоб процесор міг якомога швидше перейти в цей режим і приступити до обробки переривання. З цього випливає, що інші джерела переривань повинні бути підключені до лінії переривання IRQ. У простих системах дані джерела можна об'єднати за допомогою багатовходового елемента «АБО». У цьому випадку при виникненні переривання процесор повинен перевірити кожний периферійний пристрій для визначення джерела переривання, що може зайняти досить багато часу. Очевидно, що нам потрібно більш складне рішення.

Для більш ефективної обробки зовнішніх переривань в мікроконтролери було додано спеціальний модуль (рисунок 3.21), який називають векторним контролером переривань (VectorInterruptController–VIC).

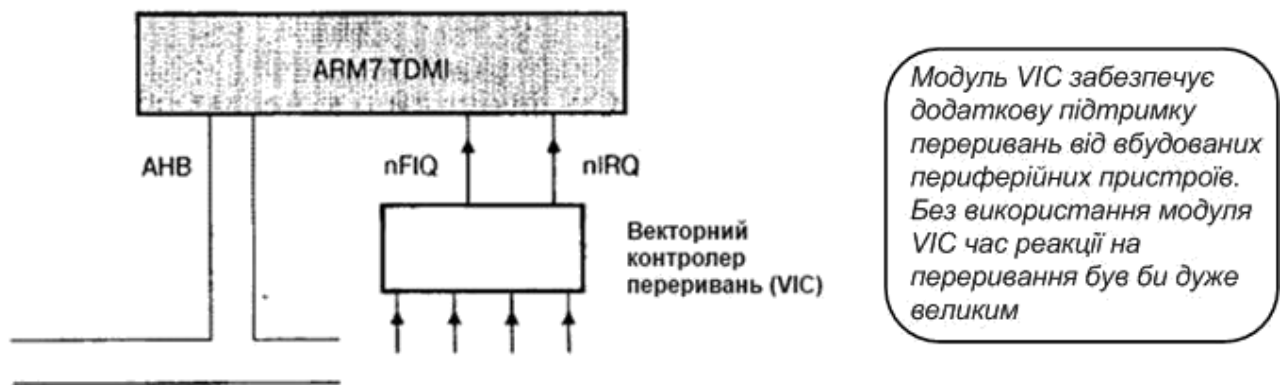


Рисунок 3.21 – Підтримка переривань за допомогою модуля VIC

Модуль VIC являє собою високо оптимізований контролер переривань. Модуль VIC використовується для обробки всіх джерел переривань від вбудованих периферійних пристроїв. Кожен з внутрішніх джерел переривань підключено до певного каналу модуля VIC. У програмі кожен з цих каналів можна підключити до ліній переривань ЦПП (FIQ і IRQ) одним із трьох способів.

Кожне переривання може оброблятися як швидке переривання FIQ, векторне переривання IRQ або ж не векторне переривання IRQ. Всі три типи переривань мають різний час відгуку. Найшвидшим є переривання FIQ, далі за ним йде векторне переривання, а найповільніше процесор реагує на неекторні переривання IRQ. Нижче розглядаються ці три методи обробки переривань. Регістри модуля VIC зображено на рисунку 3.22.

#### 4.1. Переривання FIQ

Будь яке джерело переривання може використовуватися в якості переривання FIQ. У регістрі вибору типу переривання модуля VIC (регістр VICIntSelect) кожному джерелу переривань відповідає свій біт. При встановленні цього біта обраний канал контролера підключається до лінії переривання FIQ. В ідеальному випадку у нас має бути тільки одне переривання FIQ.

Адреса	Тип	Розмір	Значення скидання	Назва
VIC base +0x000	Read	32	0x00000000	VICIRQSTATUS
VIC base +0x004	Read	32	0x00000000	VICFIQSTATUS
VIC base +0x008	Read	32	-	VICRAWINTR
VIC base +0x00C	Read/write	32	0x00000000	VICINTSELECT
VIC base +0x0010	Read/write	32	0x00000000	VICINTENABLE
VIC base +0x0014	Write	32	-	VICINTENCLEAR
VIC base +0x0018	Read/write	32	0x00000000	VICSOFTINT
VIC base +0x001C	Write	32	-	VICSOFTINTCLEAR
VIC base +0x0020	Read/write	1	0x0	VICPROTECTION
VIC base +0x0024	Read/write	16	0xFFFF	VICSWPRIORITY MASK
VIC base +0x0028	Read/write	4	0xF	VICPRIORITYDAISY
VIC base +0x100	Read/write	32	0x00000000	VICVECTADDR0
VIC base +0x104	Read/write	32	0x00000000	VICVECTADDR1
* * *				
VIC base +0x17C	Read/write	32	0x00000000	VICVECTADDR31
VIC base +0x200	Read/write	4	0xF	VICVECTPRIORITY0
VIC base +0x204	Read/write	4	0xF	VICVECTPRIORITY1
* * *				
VIC base +0x27C	Read/write	4	0xF	VICVECTPRIORITY31
VIC base +0xF00	Read/write	32	0x00000000	VICADDRESS
VIC base +0xFE0	Read	8	0x92	VICPERIPHID0
VIC base +0xFE4	Read	8	0x11	VICPERIPHID1

Рисунок 3.22 – Адреси регістрів модуля VIC

Тим не менш, при одночасному встановленні декількох бітів вказаного регістра буде дозволено кілька джерел переривань FIQ. У цьому випадку за допомогою регістра стану FIQ модуля VIC (VICFIQStatus) можна буде при вході

в обробник визначити конкретне джерело переривання і перейти до необхідної секції обробника.

Очевидно, що наявність декількох джерел переривань FIQ уповільнює перехід до процедури обробки переривань. Після призначення джерела переривання FIQ, переривання можна дозволити, використовуючи регістр дозволу переривань модуля VIC (VICIntEnable). Крім того, необхідно сконфігурувати всі периферійні пристрої, що генерують переривання, а також дозволити їм генерацію цих переривань. При генерації переривання FIQ процесор переходить в режим FIQ, а програма переходить за адресою 0x0000001C – вектору переривання FIQ. Для обслуговування переривання треба розмістити в цьому місці команду переходу до потрібної процедури обробки переривання.

### Вихід із переривання FIQ

Оголошення Cі-функції в якості оброблювача переривання FIQ вказує компілятору використовувати коректну команду повернення для відновлення виконання фоновому коду з того місця, де його виконання було перерване. Однак перш ніж вийти з процедури обробки переривання, потрібно переконатися, що прапорці переривань всіх периферійних пристроїв скинуто (рисунок 3.23).



Рисунок 3.23 – Скидання прапорця переривання

В іншому випадку переривання буде генеруватися постійно, доки відповідний прапорець не буде скинуто. Для скидання прапорця треба записати у відповідний йому біт регістра не 0, а 1.

## 4.2. Векторні переривання IRQ

Якщо одне з джерел переривань визначено як переривання FIQ, то інші джерела мають бути підключені до лінії IRQ. Для швидкої та ефективної обробки цих переривань в модулі VIC мається апаратно реалізована

програмована таблиця, в якій зберігаються адреси всіх підпрограм обробки переривань. У модулі VIC є 32 слоти для векторної адресації переривань. Для кожного слоту передбачено два регістри – регістр адреси вектора (VICVectAddr) і регістр керування вектором (VICVectCntl).

Регістри пріоритету вектора (VICVectPriorityxx) дозволяють призначати кожному слоту переривань свій пріоритет. У загальній складності підтримується 16 рівнів пріоритету, від 15 (найменший пріоритет) до 0 (найбільший пріоритет). Після скидання всі слоти переривань мають мінімальний рівень пріоритету, який може бути збільшений користувачем. Якщо двом слотам буде призначений один і той же пріоритет, то при виникненні переривань від обох джерел першим буде обслугований слот з найменшим номером.

Другим регістром для керування слотом в модулі VIC є регістр адреси вектора (VICVectAddr). Як випливає з його назви, в цьому регістрі повинна зберігатися адреса відповідної Сі-функції, яка буде запускатися при виникненні переривання, пов'язаного з даним слотом. Насправді, при генерації векторного переривання канал переривання підключається до заданого йому слоту, а адреса підпрограми обробки переривання, що знаходиться в регістрі адреси вектора даного слоту, завантажується в окремий регістр адреси вектора (рисунки 3.24). Таким чином, при генерації переривання, яке сконфігуровано як векторне, адреса його оброблювача буде завантажена в фіксовану комірку пам'яті, яка називається регістром адреси вектора (VICVectAddrxx).

Поки модуль VIC виконує всі ці дії, процесор ARM7 перемикається в режим IRQ і переходить за адресою вектора переривання IRQ (0x00000018). Щоб виконати відповідну підпрограму обробки переривання, необхідно завантажити в РС адресу з регістра адреси вектора модуля VIC. Наступна команда асемблера виконує цю операцію за один такт:

```
LDR    PC, [PC, #-0x0120].
```

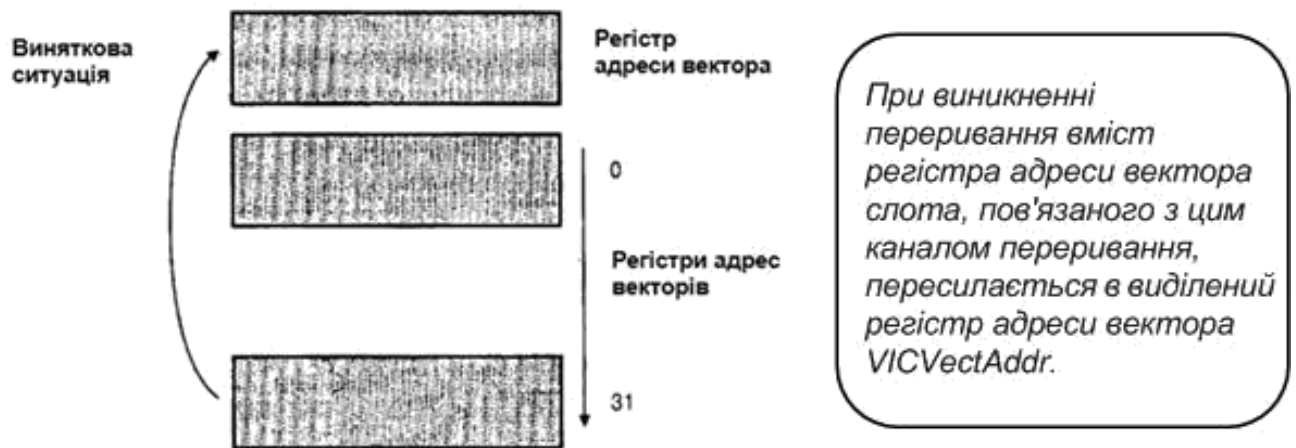


Рисунок 3.24 – Отримання адреси векторного переривання IRQ

Оскільки ми вже знаходимося в режимі IRQ, то нам відомо, що поточна адреса дорівнює  $0x00000018 + 8$  (з урахуванням конвеєра). При відніманні з цього значення числа  $0x0120$  відбудеться перехід через верхню межу 32-бітного адресного простору, в результаті чого вказана команда завантажить в лічильник команд вміст комірки пам'яті з адресою  $0xFFFFFFFF00$  (вміст регістра адреси вектора), рисунок 3.25.

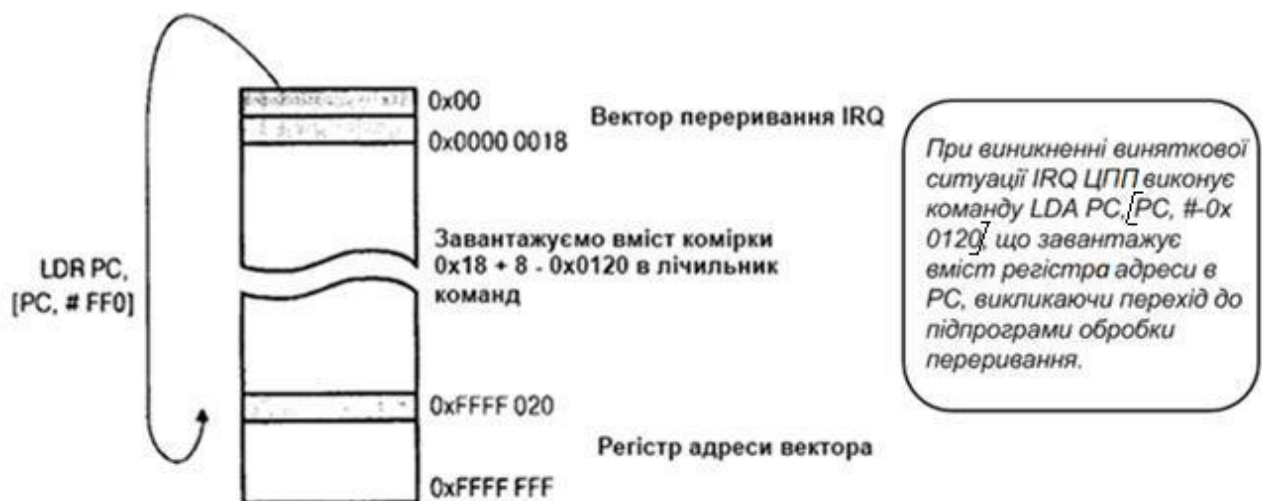


Рисунок 3.25 – Обробка векторного переривання IRQ

Як і у випадку переривання FIQ, потрібно переконатися в тому, що всі прапорці переривань периферійного пристрою, які генерують запит на переривання, скинуті. Крім того, в кінці обробника переривання потрібно виконати холосту операцію запису в регістр адреси вектора VICVectAddr



(рисунок 3.26). Ця операція вказує модулю VIC на закінчення обробки переривання і дозволяє оброблення відкладених переривань IRQ.

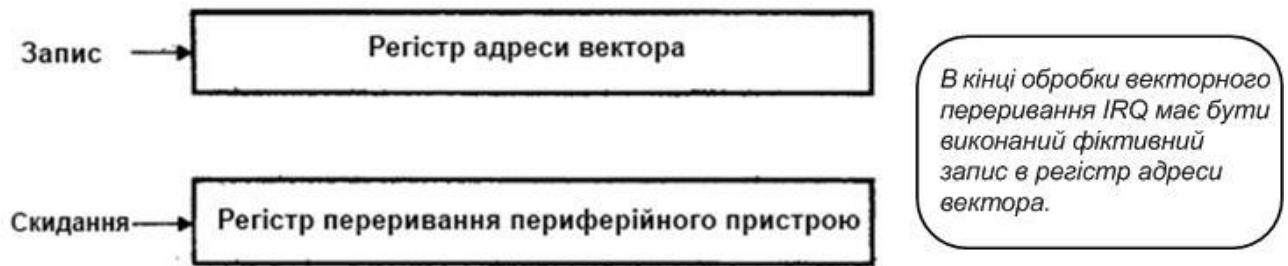


Рисунок 3.26 – Закінчення обробки векторного переривання IRQ

## 5. Невекторні переривання

Модуль VIC може обробляти до 32 векторних переривань IRQ і принаймні одне швидке переривання FIQ. Якщо в мікроконтролері є більше 33 джерел переривань, то решта може обслуговуватися як неекторні переривання. Всі джерела неекторних переривань обслуговуються однією єдиною підпрограмою обробки переривань. Адреса цієї підпрограми зберігається в додатковому регістрі адреси вектора, що називається регістром адреси вектора за замовчуванням (VICDefVectAddr). Якщо в модулі VIC переривання дозволено і при цьому не сконфігуровано як переривання FIQ, то воно буде сприйнято як неекторне переривання. При генерації такого переривання адреса із регістра VICDefVectAddr перезавантажується в регістр адреси вектора VICVectAddr, що викликає перехід процесора до цієї підпрограми. При вході в цю підпрограму ЦПП повинен прочитати регістр статусу IRQ, щоб визначити, яке із джерел неекторних переривань згенерувало його.

## 6. Програмні переривання

Модуль VIC надає можливість прикладному ПЗ генерувати переривання по кожному з каналів, використовуючи регістри програмного переривання модуля (рисунок 3.27). Ці регістри не мають ніякого відношення до команди програмного переривання (SWI), однак дозволяють перевірити джерела переривань (при включенні живлення або ж при налагодженні під час розробки програми).

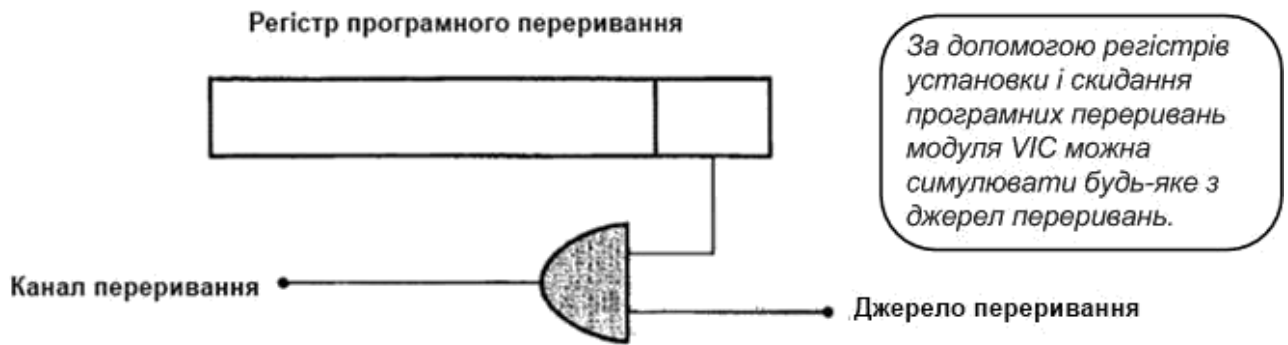


Рисунок 3.27 – Регістр програмного переривання

Крім усього іншого, в модулі VIC передбачено спеціальний захищений режим, що запобігає доступ до будь якого реєстра модуля в режимі User. Якщо прикладній програмі потрібен доступ до модуля VIC, то вона повинна перемкнутися в привілейований режим. Для цього можна використовувати переривання FIQ або IRQ, а також команду SWI.

Типові значення затримок для різних переривань наведено у [4]. У разі неекторних переривань затримка складається з часу реакції на векторне переривання і часу, необхідного для зчитування реєстра стану IRQ і визначення джерела.

## 7. Вкладені переривання

На апаратному рівні ЦПП ARM7 і модуль VIC не підтримують вкладені переривання. Якщо у вашій програмі вимагається можливість переривати виконання підпрограми обробки переривань, це можна зробити програмно за допомогою пари макросів (рисунок 3.28). Перш ніж перейти до розгляду вкладених переривань, треба звернути увагу на те, що при обробці процесором ARM7 зовнішнього переривання, переривання IRQ забороняються. Крім того, при вході в Si-функцію, оголошену як обробник переривання IRQ, реєстр LR\_isr зберігається в стеку.

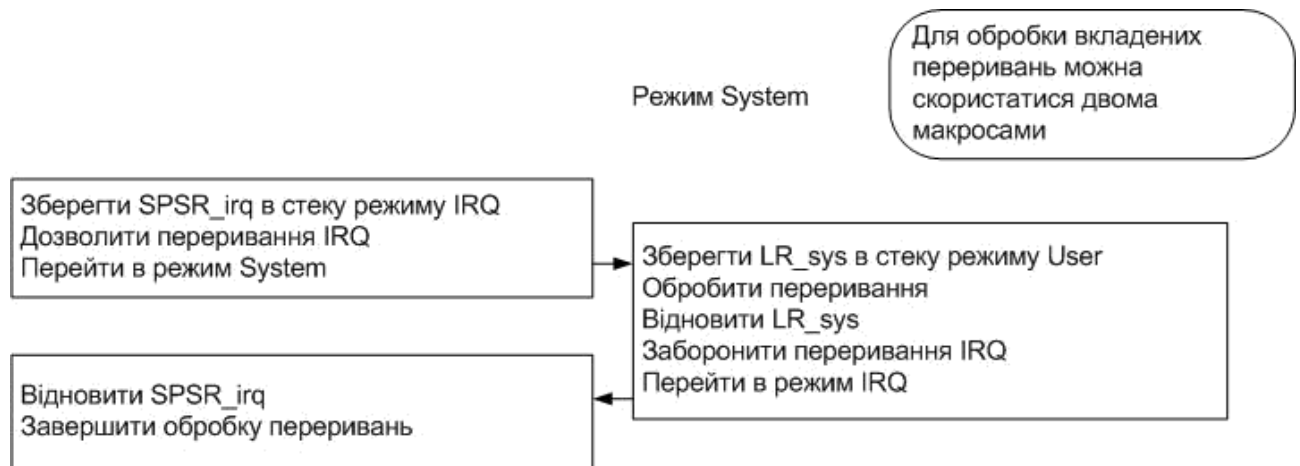


Рисунок 3.28 – Обробка вкладених переривань

Після того як процесор увійшов в підпрограму обробки переривання IRQ, нам потрібно виконати ряд команд для включення підтримки вкладених переривань. Перш за все, необхідно зберегти регістр SPSR\_irq в стеку. Це дозволить нам коректно відновити стан регістра CPSR при поверненні в режим User. Потім треба дозволити переривання IRQ і перейти в режим System (цей режим аналогічний режиму User, проте в ньому можна використовувати команди MSR та MRS). У режимі System необхідно зберегти новий регістр зв'язку, тому що в ньому може зберігатися значення, яке використовується в фоні (в режимі User). Тому даний регістр заноситься в стек режиму System. Після виконання цих операцій ми можемо приступити до виконання підпрограми обробки переривання. Після завершення обробки запускається макрос, що виконує зворотні операції. Цей макрос відновлює стан регістра зв'язку, забороняє переривання IRQ, перемикає процесор в режим IRQ, відновлює регістр SPSR\_irq, після чого обробка переривань вважається завершеною. Код описаних макросів наведено нижче.

```

#define IENABLE
    __asm { MRS          LR, SPSR }
    __asm { STMFD SP!, {LR} }
    __asm { MSR          CPSR_c, #0x1F }
    __asm { STMFD SP', {LR} }

#define IDISABLE
  
```

```

__asm { LDMFD SR!, {LR} }
__asm { MSR          CPSR_c, #0x92 }
__asm { LDMFD SP!, {LR} }
__asm { MSR          SPSR_cxsf, LR }

```

Цей код складається з восьми команд (32 байти коду ARM), а загальний час виконання обох макросів становить 230 нс. Описана методика дозволяє реалізувати обробку відповідного переривання іншим перериванням. Якщо вам необхідно задати пріоритети вкладених переривань, то в макросі слід передбачити блокування переривань з низьким пріоритетом шляхом заборони в модулі VIC джерел переривань з більш низьким пріоритетом.

### 8. Джерела переривань

В таблиці 3.10 наведено список джерел переривань для окремих периферійних пристроїв.

Таблиця 3.10 – Джерела переривань

Блок	Прапорці	Канал VIC
WDT	Переривання вартового таймера (WDINT)	0
–	Зарезервовано для програмних переривань	1
Ядро ARM	Вбудований ICE DbgCommRx	2
Ядро ARM	Вбудований ICE DbgCommTx	3
TIMER0	Співпадіння 0..3 (MR0, MR1, MR2, MR3)	4
	Захоплення 0...3 (CR0, CR1, CR2, CR3)	
TIMER1	Співпадіння 0..3 (MR0, MR1, MR2, MR3)	5
	Захоплення 0...3 (CR0, CR1, CR2, CR3)	
UART0	Стан лінії Rx (RLS)	6
	Регістр передачі пустий (THRE)	
	Прийняття даних доступу (RDA)	
	Індикатор тайм-ауту (CTI)	

## Продовження таблиці 3.10

Блок	Прапорці	Канал VIC
UART1	Стан лінії Rx (RLS)	7
	Регістр передачі пустий (THRE)	
	Прийняття даних доступу (RDA)	
	Індикатор тайм-ауту (CTI)	
	Переривання за станом модему	
PWM0	Співпадіння 0...6 (MR0, MR1, MR2, MR3, MR4, MR5, MR6)	8
I2C	SI (зміна стану)	9
SPI0	Прапорець переривання від SPI0 (SPIF)	10
	Режим помилки (MODF)	
SPI1	Прапорець переривання від SPI1 (SPIF)	11
	Режим помилки (MODF)	
PLL	Петлю ФАПЧ замкнуто (PLOCK)	12
RTC	Лічильник інкрементування (RTCCIF)	13
	Тривога (RTCALF)	
Керування системою	Зовнішнє переривання 0 (EINT0)	14
Керування системою	Зовнішнє переривання 1 (EINT1)	15
Керування системою	Зовнішнє переривання 2 (EINT2)	16
Керування системою	Зовнішнє переривання 3 (EINT3)	17
A/D	АЦП	18

## Продовження таблиці 3.10

Блок	Прапорці	Канал VIC
CAN	CAN1 Tx	20
	CAN2 Tx	21
	CAN3 Tx (лише у LPC2194/2292/2294	22
	інакше зарезервовано)	
	CAN4 Tx (лише у LPC2194/2292/2294	23
	інакше зарезервовано)	
	Зарезервовано	24,25
	CAN1 Rx	26
	CAN2 Rx	27
CAN	CAN3 Rx (лише у LPC2194/2292/2294	28
	інакше зарезервовано)	
	CAN3 Rx (лише у LPC2194/2292/2294	29
	інакше зарезервовано)	
	Зарезервовано	30,31

## ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) Яку роль виконують переривання в мікропроцесорних системах?
- 2) Що відбувається в роботі мікропроцесорної системи при виникненні переривання?
- 3) Назвіть два види переривань в мікропроцесорних системах.
- 4) Чим відрізняються апаратне та програмне переривання?
- 5) Чим відрізняються переривання загального призначення (IRQ) і швидке переривання (FIQ)?
- 6) Поясніть призначення блоку вибору функції виводів.
- 7) За допомогою яких регістрів здійснюється керування зовнішніми перериваннями?
- 8) Який модуль в мікроконтролерах ARM7 використовується для більш ефективної обробки зовнішніх переривань?
- 9) Для обробки яких джерел переривань використовується модуль VIC?
- 10) Як в мікроконтролері програмується переривання FIQ?
- 11) Як виконується вихід із переривання FIQ?
- 12) Опишіть особливості обробки векторних переривань IRQ.
- 13) Яку адресу має вектор переривання IRQ?
- 14) Як виконується закінчення обробки векторного переривання IRQ?
- 15) Які переривання можуть обслуговуватися як неекторні?
- 16) Якою підпрограмою обслуговуються джерела неекторних переривань?
- 17) Як модуль VIC надає можливість прикладному ПЗ генерувати переривання?
- 18) Як в мікроконтролері підтримується обробка вкладених переривань?
- 19) Назвіть джерела переривань від окремих периферійних пристроїв.
- 20) Де знаходиться підпрограма, яка викликається за перериванням?
- 21) Наведіть приклади подій, які можуть ініціювати зовнішнє переривання.

# Модуль прямого доступу до пам'яті

## 1. Загальна характеристика

Контролер прямого доступу до пам'яті (Direct Memory Access – DMA) є представником сімейства модулів PrimeCell™, розроблених компанією ARM, і оптимізований для роботи спільно з шинною структурою ядра ARM [1, 4, 7]. Контролер DMA загального призначення підключено до шини АНВ через два порти – ведений, через який ЦПП може звертатися до регістрів контролера, і ведучий, за допомогою якого механізм прямого доступу до пам'яті здійснює захоплення шини, а також виконує її арбітраж з ЦПП ARM7 і спеціалізованими модулями DMA, контролерами USB і Ethernet.

У складі контролера DMA є два незалежних модуля прямого доступу до пам'яті, кожен з яких може бути налаштований для здійснення пересилання даних з пам'яті в пам'ять, з пам'яті до периферійного пристрою, з периферійного пристрою в пам'ять, а також з одного периферійного пристрою в інший (рисунок 3.29). По закінченню пересилання контролер DMA може згенерувати переривання, при цьому сигнали переривань від усіх модулів DMA об'єднуються за схемою АБО і підключаються до одного – єдиного каналу переривань модуля VIC.

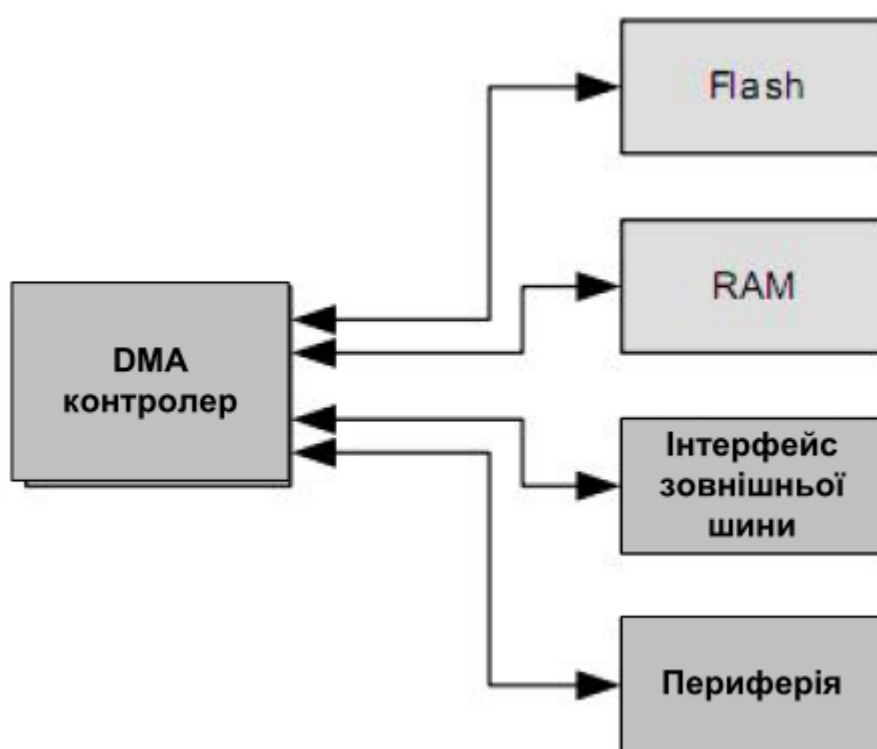


Рисунок 3.29 – Види пересилання даних контролером DMA



## 2. Огляд модуля DMA

Для того, щоб вивчити роботу модуля DMA, розглянемо, для початку, найпростішу операцію – пересилання даних з пам'яті в пам'ять. Для виконання цієї операції модуль DMA спочатку виграє арбітраж на шині АНВ, після чого копіює дані з джерела у свій внутрішній FIFO–буфер. Потім вміст внутрішнього буфера записується в задану область пам'яті. Зчитування і запис модулем DMA даних може здійснюватися як у вигляді одноразових пересилань, так і у вигляді пакетів з декількох пересилань. В останньому випадку модуль DMA може блокувати внутрішні шини до завершення кожного етапу пересилання. Також модуль DMA може зчитувати і записувати блоки даних різного розміру. Це означає, зокрема, що в процесі операції пересилання може здійснюватися упакування і розпакування даних. Наприклад, можна зчитати з пам'яті чотири 32 – бітних слова даних, а потім пересилати їх в буфер передавача UART по 16 біт. Після того, як модуль DMA виграє арбітраж на шині і буде готовий до пересилання даних, він дозволить керувати процесами зчитування і запису цих даних. Однак у разі пересилання даних між пам'яттю і периферійними пристроями у якості контролера обміну може виступити саме периферійний пристрій, що дозволяє запис або зчитування даних по мірі готовності. Крім того, модуль DMA підтримує пересилання несуміжних даних. Для кожного з модулів ви можете у вигляді пов'язаного списку задати послідовність пересилань, які згодом будуть автоматично виконуватися один за одним. Це дозволяє зчитувати дані з несуміжних областей пам'яті, а потім і передавати їх у вигляді єдиного блоку в пам'ять або периферійний пристрій. Аналогічно можна запрограмувати модуль DMA таким чином, щоб він «розкидав» вміст єдиного блоку пам'яті за різними адресами.

Всього у контролері DMA є 24 регістри, які можна розбити на три групи: 14 регістрів стану і конфігурації та по 5 регістрів керування кожного з модулів. Загальні регістри стану і конфігурації (рисунки 3.30) відповідають, в основному, за дозвіл роботи контролера DMA і керування індивідуальними перериваннями модулів.

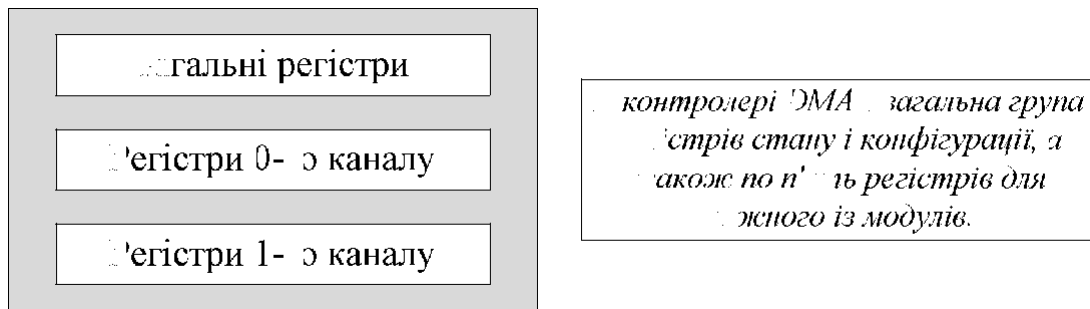


Рисунок 3.30 – Групи регістрів контролера DMA

Дозвіл роботи контролера DMA здійснюється встановленням біта E регістра конфігурації DMACConfiguration. Кожний з модулів DMA має два переривання – переривання terminal count, що генерується при завершенні пересилання, і переривання помилки, яке генерується при виявленні модулем помилки на шині. Дозволи для обох джерел переривань здійснюються в регістрі конфігурації каналу DMACCOConfiguration і DMACCIConfiguration. Крім того, для кожного джерела переривання мається по парі регістрів стану: регістр стану переривання і регістр необробленого стану переривання. Останній регістр відображає стан усіх прапорців переривань, незалежно від того, було відповідне переривання дозволено чи ні, тоді як звичайний регістр стану переривання відображає стан тільки дозволених переривань DMA. Для ініціювання одиночного або групового пересилання за DMA, у разі, якщо модуль DMA виступає в ролі контролера потоку, використовуються регістри програмного запиту на одиночне (DMACSoftSReq) і пакетне пересилання (DMACSoftBReq) відповідно.

### 3. Синхронізація DMA

Модулі DMA можуть здійснювати пересилання через всі внутрішні шини мікроконтролерів LPC2300/2400. Якщо ці шини працюють з різними швидкостями, необхідно в регістрі синхронізації DMACSync встановити в 1 біти синхронізації для різних запитів DMA. Це дозволить уникнути виникнення конфліктів на шинах, але й збільшить час відгуку модулів.

### 4. Пересилання з пам'яті в пам'ять

Після дозволу роботи модуля DMA і конфігурування переривань можна приступати до конфігурування власне регістрів каналів. При здійсненні пересилання даних з пам'яті в пам'ять початкові адреси джерела і приймача

заносяться в однойменні регістри (DMACCxSrcAddr і DMACCxDcstAddr). Регістр елемента зв'язаного списку використовується при пересиланні несуміжних даних і для одиночного пересилання повинен бути скинутий в 0.

У регістрі керування (рисунок 3.31) можна задати розмір пересилання, у разі якщо роль контролера потоку грає модуль DMA.

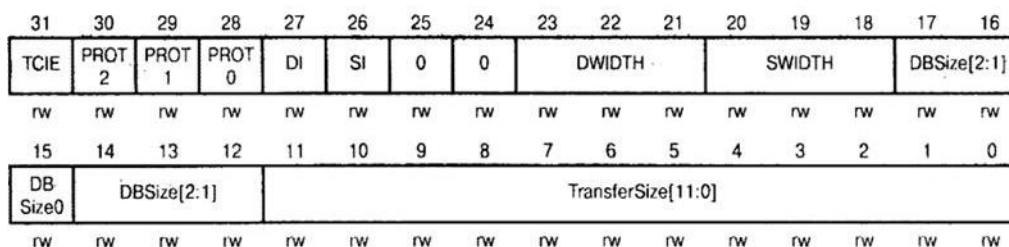


Рисунок 3.31 – Регістр керування модуля DMA

Якщо ж в якості контролера потоку виступає периферійний пристрій, то в цьому полі регістра повинен бути записаний 0. У процесі пересилання вміст даного поля декрементується, однак, щоб прочитати його і отримати осмислене значення, вам необхідно спочатку заборонити роботу модуля DMA. Поля розрядності джерела та приймача дозволяють задати розмір слів, що зчитуються і записуються модулем DMA. При цьому можна задавати різні значення вказаних полів. Упакування та розпакування даних буде здійснюватися модулями автоматично. При необхідності адреси джерела і приймача можуть інкрементуватися після кожного пересилання, для чого слід встановити біти SI і DI відповідно. Завдяки цьому ви зможете переслати блок даних з однієї безперервної області пам'яті в іншу або ж скопіювати блок даних в якусь область пам'яті, наприклад, групу регістрів периферійного пристрою. У цьому ж регістрі можна задати різні опції захисту. Встановивши біт PROT0, можна заборонити на час пересилання звернення до регістрів DMA з коду, що виконується в режимі User. Біт PROT1 дозволяє включити буферизацію. При цьому пересилання даних буде здійснюватися з максимально можливою швидкістю, проте може відбутися порушення однорідності даних, оскільки буферизовані дані повинні бути записані в ОЗП. Якщо встановити біт дозволу переривання термінального рахунку, то в кінці кожного DMA – пересилання буде генеруватися переривання, яке повідомить ЦПП про те, що модуль DMA готовий до подальшого використання.

Останнє поле регістра керування дозволяє задати розмір пакетного пересилання.

## 5. Пакетна передача

Кожен з модулів DMA здатний зчитувати одиночне слово і передавати його за заданою адресою. Перш ніж модуль DMA зможе виступити в якості майстра шини, він повинен на час цих операцій відібрати шину у ЦПП і інших модулів DMA. Сконфігурувавши відповідним чином поля розміру пакета джерела і приймача в регістрі керування, можна здійснювати пересилання декількох слів даних в пакетному режимі. При цьому кожен з модулів DMA дозволяє здійснювати в пакетному режимі до 256 пересилань. Встановивши біт L регістра конфігурації каналу (DMACCxConfiguration), можна попередити звільнення шини модулем DMA до завершення пересилання.

## 6. Підтримка модулем DMA периферійних пристроїв

У ролі контролера потоку для кожного з модулів DMA можуть виступати периферійні пристрої.

Необхідно дозволити використання модуля DMA конкретним периферійним пристроєм і записати в поля пристрою – джерела і пристрою – приймача регістра конфігурації каналу DMA (DMACCxConfiguration) значення, відповідні використовуваному пристрою (рисунок 3.32).

Після дозволу роботи модуля DMA він буде керуватися відповідним периферійним пристроєм.



Рисунок 3.32 – Регістр конфігурації модуля DMA

## 7. Пересилання несуміжних даних

Обидва модулі DMA підтримують пересилання несуміжних даних. Цей механізм дозволяє запрограмувати виконання декількох DMA – пересилань в заданій послідовності, в результаті чого можна буде зібрати дані, які розташовано в різних областях пам'яті, і помістити їх в одну безперервну область. Точно також за допомогою декількох DMA – пересилань, що виконуються автоматично, можна «розкидати» дані з однієї області пам'яті за різними адресами. Області пам'яті в ОЗП задаються у вигляді «елементів» списку, що представляють собою запис, в якому міститься адреса джерела, адреса приймача, адреса наступного елемента списку і керуюче слово. Початкова адреса такого елемента заноситься в регістр елемента зв'язаного списку модуля DMA, і після закінчення поточного пересилання елемент, на який посилається поточний, буде автоматично завантажений у регістр керування каналу DMA. При цьому регістр елемента зв'язаного списку буде вказувати на наступний елемент. За рахунок такого механізму зв'язаних списків ми можемо зв'язати кілька DMA – пересилань. Для останнього пересилання в ланцюжку необхідно в регістрі керування дозволити переривання DMA з тим, щоб після його завершення було згенеровано переривання і можна було б ініціалізувати нові DMA – пересилання.

## 8. Регістри керування модулем DMA

Регістри керування модулем DMA наведено у таблиці 3.11.

Таблиця 3.11 – Регістри керування модулем DMA

Назва	Опис	Доступ	Початкове значення	Адреса
<b>Загальні регістри</b>				
DMACIntStatus	Interrupt Status Register	RO	0x0	0xFFE0 4000
DMACIntTCStatus	Interrupt Terminal Count Status Register	RO	0x0	0xFFE0 4004
DMACIntTCClear	Interrupt Terminal Count Clear Register	WO	–	0xFFE0 4008
DMACIntErrorStatus	Interrupt Error Status Register	RO	0x0	0xFFE0 400C

Продовження таблиці 3.11

Назва	Опис	Доступ	Початкове значення	Адреса
DMACIntErrClr	Interrupt Error Clear Register	WO	-	0xFFE0 4010
DMACRawIntTCStatus	Raw Interrupt Terminal Count Status Register	RO	-	0xFFE0 4014
DMACRawIntErrorStatus	Raw Error Interrupt Status Register	RO	-	0xFFE0 4018
DMACEnbldChns	Enabled Channel Register	RO	0x0	0xFFE0 401C
DMACSoftBReq	Software Burst Request Register	R/W	0x0000	0xFFE0 4020
DMACSoftSReq	Software Single Request Register	R/W	0x0000	0xFFE0 4024
DMACSoftLBReq	Software Last Burst Request Register	R/W	0x0000	0xFFE0 4028
DMACSoftLSReq	Software Last Single Request Register	R/W	0x0000	0xFFE0 402C
DMACConfiguration	Configuration Register	R/W	0x0000 0000	0xFFE0 4030
DMACSync	Synchronization Register	R/W	0x0000	0xFFE0 4034
<b>Регістри Channel 0</b>				
DMACC0SrcAddr	Channel 0 Source Address Register	R/W	0x0000 0000	0xFFE0 4100
DMACC0DestAddr	Channel 0 Destination Address Register	R/W	0x0000 0000	0xFFE0 4104
DMACC0LLI	Channel 0 Linked List Item Register	R/W	0x0000 0000	0xFFE0 4108
DMACC0Control	Channel 0 Control Register	R/W	0x0000 0000	0xFFE0 410C
DMACC0Configuration	Channel 0 Configuration Register	R/W	0x00000 [1]	0xFFE0 4110
<b>Регістри Channel 0</b>				
DMACC1SrcAddr	Channel 1 Source Address Register	R/W	0x0000 0000	0xFFE0 4120
DMACC1DestAddr	Channel 1 Destination Address Register	R/W	0x0000 0000	0xFFE0 4124
DMACC1LLI	Channel 1 Linked List Item Register	R/W	0x0000 0000	0xFFE0 4128
DMACC1Control	Channel 1 Control Register	R/W	0x0000 0000	0xFFE0 412C
DMACC1Configuration	Channel 1 Configuration Register	R/W	0x00000 [1]	0xFFE0 4130

1 – біт [17] тільки для читання

## 8.1. Загальні регістри

### 8.1.1 Регістр статусу переривань (Interrupt Status Register, DMACIntStatus – 0xFFE0 4000)

Регістр DMACIntStatus (таблиця 3.12) доступний тільки для читання і відображає стан переривань після маскуванню. Високий рівень позначає активний запит конкретного DMA каналу на переривання. Запит може бути викликаний або через помилку, або через запити на переривання термінального рахунку.

Таблиця 3.12 – Опис розрядів регістра DMACIntStatus

Біт	Символ	Опис	Початкове значення
0	IntStatus0	Стан переривань 0 каналу після маскуванню	0
1	IntStatus1	Стан переривань 1 каналу після маскуванню	0
31:2	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів невизначений.	N/A

### 8.1.2 Регістр статусу переривань термінального рахунку (Interrupt Terminal Count Status Register, DMACIntTCStatus – 0xFFE0 4004)

Регістр DMACIntTCStatus (таблиця 3.13) доступний тільки для читання і відображає стан термінального рахунку після маскуванню. Таблиця 3.13 – Опис розрядів регістра DMACIntTCStatus

Біт	Символ	Опис	Початкове значення
0	IntTCStatus0	Стан запиту термінального рахунку на переривання 0 каналу -	0
1	IntTCStatus1	Стан запиту термінального рахунку на переривання 1 каналу -	0
31:2	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів невизначений.	N/A

### 8.1.3 Регістр скидання переривання термінального рахунку (Interrupt Terminal Count Clear Register, DMACIntClear – 0xFFE0 4008)

Регістр DMACIntTCClear (таблиця 3.14) доступний тільки для запису та скидає запит термінального рахунку на переривання. При записі до регістра, кожен біт даних високого рівня скидає відповідний біт регістра статусу, біти низького рівня не впливають на відповідні біти регістра.

Таблиця 3.14 – Опис розрядів регістра DMACIntTCClear

Біт	Символ	Опис	Початкове значення
0	IntTCClear0	Запис 1 скидає запит термінального рахунку на переривання для 0 каналу	–
1	IntTCClear1	Запис 1 скидає запит термінального рахунку на переривання для 1 каналу	–
31:2	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів невизначений.	N/A

#### 8.1.4 Регістр статусу переривання за помилкою (Interrupt Error Status Register, DMACIntErrorStatus – 0xFFE0 400C)

Регістр DMACIntErrorStatus (таблиця 3.15) доступний для читання і відображає стан помилки переривання після маскуваня.

Таблиця 3.15 – Опис розрядів регістра DMACIntErrorStatus

Біт	Символ	Опис	Початкове значення
0	IntErrorStatus0	Стан переривання за помилкою для каналу 0	0x0
1	IntErrorStatus1	Стан переривання за помилкою для каналу 1	0x0
31:2	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів невизначений.	N/A

#### 8.1.5 Регістр скидання переривання за помилкою (Interrupt Error Clear Register, DMACIntErrClr – 0xFFE0 4010)

Регістр DMACIntErrClr (таблиця 3.16) доступний тільки для запису і скидає запити на переривання «помилки». При запису до регістра, кожен біт даних високого рівня скидає відповідний біт регістра статусу, біти низького рівня не впливають на відповідні біти регістра. Таблиця 3.16 – Опис розрядів регістра DMACIntErrClr

Біт	Символ	Опис	Початкове значення
0	IntErrClr0	Запис 1 скидає запит на переривання «помилки» для 0 каналу (IntErrorStatus0)	–
1	IntErrClr1	Запис 1 скидає запит на переривання «помилки» для 1 каналу (IntErrorStatus1)	–
31:2	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів невизначений.	N/A



### 3.9.8.1.6 Регістр статусу необробленого переривання термінального Рахунку (Raw Interrupt Terminal Count Status Register, DMACRawIntTCStatus – 0xFFE0 4014)

Регістр DMACRawIntTCStatus (таблиця 3.17) доступний тільки для читання і відображає стан каналу, який надіслав запит на закінчення передачі (переривання термінального рахунку) до маскуванню. Високий рівень біта відображає активне переривання термінального рахунку до маскуванню. Таблиця 3.17 – Опис розрядів регістра DMACRawIntTCStatus

Біт	Символ	Опис	Початкове значення
0	RawIntTCStatus0	Стан переривання термінального рахунку для каналу 0 до маскуванню.	–
1	RawIntTCStatus1	Стан переривання термінального рахунку для каналу 1 до маскуванню.	–
31:2	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів невизначений.	N/A

### 3.9.8.1.7 Регістр статусу необробленого переривання за помилкою (Raw Error Interrupt Status Register, DMACRawIntErrorStatus–0xFFE0 4018)

Регістр DMACRawIntErrorStatus (таблиця 3.18) доступний тільки для читання і відображає стан каналу, який надіслав запит на переривання «помилки» до маскуванню. Високий рівень біта відображає активний запит на переривання «помилки» до маскуванню.

Таблиця 3.18 – Опис розрядів регістра DMACRawIntErrorStatus

Біт	Символ	Опис	Початкове значення
0	RawIntErrorStatus0	Стан переривання «помилки» для каналу 0 до маскуванню	–
1	RawIntErrorStatus1	Стан переривання «помилки» для каналу 1 до маскуванню	–
31:2	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів невизначений.	N/A

### 3.9.8.1.8 Регістр увімкнення каналу (Enabled Channel Register, DMACEnbldChns – 0xFFE0 401C)

Регістр DMACEnbldChns (таблиця 3.19) доступний тільки для читання і відображає увімкнені канали, відповідно до біта Enable у регістрі DMACSSxConfiguration. Високий рівень біта означає увімкнений DMA – канал. Біт скидається за закінченням DMA – передачі. Таблиця 3.19 – Опис розрядів регістра DMACEnbldChns

Біт	Символ	Опис	Початкове значення
0	EnabledChannels0	Стан каналу 0	0
1	EnabledChannels1	Стан каналу 1	0
31:2	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів невизначений	N/A

### 3.9.8.1.9 Регістр запиту програми на пакетну передачу (Software Burst Request Register, DMACSoftBReq – 0xFFE0 4020)

Регістр DMACSoftBReq (таблиця 3.20) доступний для читання/запису та вмикає пакетні запити DMA, що створюються програмним забезпеченням. Запит DMA може бути згенерований для кожного джерела записом 1 до відповідного біта регістра. Біт регістра скидається за завершенням транзакції. Запис 0 до регістра нічого не змінює. Читання регістра показує джерела, які потребують пакетні передачі DMA. Запит може бути надісланий або з периферійного пристрою, або регістра.

Таблиця 3.20 – Опис розрядів регістра DMACSoftBReq

Біт	Символ	Опис	Початкове значення
0	SoftBReqSSP0Tx	Запит програми на пакетну передачу для SSP0 Tx	0
1	SoftBReqSSP0Rx	Запит програми на пакетну передачу для SSP0 Rx	0
2	SoftBReqSSP1Tx	Запит програми на пакетну передачу для SSP1 Tx	0
3	SoftBReqSSP1Rx	Запит програми на пакетну передачу для SSP1 Rx	0

Продовження таблиці 3.20

Біт	Символ	Опис	Початкове значення
4	SoftBReqSDMMC	Запит програми на пакетну передачу для SD/MMC	0
5	–	Зарезервований, запис до біта не можливий.	N/A
6	SoftBReqI2S1	Запит програми на пакетну передачу для I <sup>2</sup> S1	0
31:7	–	Зарезервовані, запис до бітів не можливий. Результат читання бітів невизначений	N/A

### 3.9.8.1.10 Регістр запиту програми на одиночну передачу (Software Single Request Register, DMACSoftSReq – 0xFFE0 4024)

Регістр DMACSoftSReq (таблиця 3.21) доступний для читання/запису і вмикає одиночні DMA – запити для програм. DMA – запит може бути згенерований для кожного джерела записом 1 до відповідного біта регістра. Біт регістра скидається за завершенням транзакції. Запис 0 до регістра нічого не змінює. Читання регістра повертає джерела, що потребують одиночних DMA передач.

Таблиця 3.21 – Опис розрядів регістра DMACSoftSReq

Біт	Символ	Опис	Початкове значення
0	SoftReqSSP0Tx	Запит програми на одиночну передачу для SSP0 Tx	0
1	SoftReqSSP0Rx	Запит програми на одиночну передачу для SSP0 Rx-	0
2	SoftReqSSP1Tx	Запит програми на одиночну пе-	0
		редачу для SSP1 Tx	
3	SoftReqSSP1Rx	Запит програми на одиночну передачу для SSP1 Rx	0
4	SoftReqSDMMC	Запит програми на одиночну передачу для SD/MMC-	0
31:5	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів не визначений	N/A

### 3.9.8.1.11 Регістр запиту програми останньої пакетної передачі (Software Last Burst Request Register, DMACSoftLBReq – 0xFFE0 4028)

Регістр DMACSoftLBReq (таблиця 3.22) доступний для запису/читання та надає можливість програмі на запит передачі останнього пакету DMA. DMA –

запит може бути згенерований для кожного джерела записом 1 до відповідного біта регістра. Біт регістра скидається за завершенням транзакції. Запис 0 до регістра нічого не змінює. Читання регістра повертає джерела, що потребують останню пакетну DMA – передачу.

Таблиця 3.22 – Опис розрядів регістра DMACSoftLBReq

Біт	Символ	Опис	Початкове значення
3:0	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів невизначений.	N/A
	SoftLBReqSDMMC	Запит програми на останню пакетну передачу для SD/MMC	0
31:5	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів невизначений	N/A

### 3.9.8.1.12 Регістр запиту програми останньої одиночної передачі (Software Last Single Request Register, DMACSoftLSReq – 0xFFE0 402C)

Регістр DMACSoftLSReq (таблиця 3.23) доступний для запису/читання та надає можливість програмі на запит останньої одиночної DMA – передачі. DMA

– запит може бути згенерований для кожного джерела записом 1 до відповідного біта регістра. Біт регістра скидається за завершенням транзакції. Запис 0 до регістра нічого не змінює. Читання регістра повертає джерела, що потребують останньої одиночної DMA – передачі.

Таблиця 3.23 – Опис розрядів регістра DMACSoftLSReq

Біт	Символ	Опис	Початкове значення
3:0	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів невизначений	N/A
4	SoftLSReqSDMMC	Запит програми на останню одиночну передачу для SD/MMC	0
31:5	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів невизначений	N/A

### 3.9.8.1.13 Регістр конфігурації (Configuration Register, DMACConfiguration – 0xFFE0 4030)

Регістр DMACConfiguration (таблиця 3.24) доступний для читання/запису та призначений для конфігурації роботи DMA. Порядок байтів АНВ – майстра може бути змінений записом відповідного значення до біта М даного регістра. Таблиця 3.24 – Опис розрядів регістра DMACConfiguration

Біт	Символ		Опис	Початкове значення
0	Е		Стан DMA:	0
		0	Відключений. DMA в режимі зниженого споживання енергії	
		1	Увімкнений	
1	М		Порядок байтів АНВ – майстра:	0
		0	Режим Little – endian	
		1	Режим Big – endian	
31:2	–	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів невизначений.	N/A

### 3.9.8.1.14 Регістр синхронізації (Synchronization Register, DMACSync – 0xFFE0 4034)

Регістр DMACSync (таблиця 3.25) доступний для читання/запису і вмикає або відключає логіку синхронізації для сигналів – запитів DMA. DMA сигнали – запити складаються з DMACBREQ [15:0], DMACSREQ [15:0], DMACLBREQ [15:0], та DMACLSREQ [15:0]. Скинутий біт вмикає логіку синхронізації для заданої групи DMA – запитів. Встановлений біт відключає логіку синхронізації для заданої групи DMA – запитів. Якщо регістр скинутий в 0, логіка синхронізації увімкнена.

### 3.9.8.2 Канальні регістри

Канальні регістри використовуються для програмування двох DMA – каналів. Ці регістри включають:

- два регістри DMACCSrcAddr;
- два регістри DMACCDestAddr;
- два регістри DMACCLLI;
- два регістри DMACControl;

– два регістри DMA<sub>CCx</sub>Configuration, де x = 0/1.

При виконанні scatter/gather – операцій DMA перші 4 регістри оновлюються автоматично

Таблиця 3.25 – Опис розрядів регістра DMA<sub>CSync</sub>

Біт	Символ	Опис	Початкове значення
15:0	DMA <sub>CSync</sub>	Увімкнення або відключення логіки синхронізації DMA. Скинутий біт вмикає логіку синхронізації для DMA <sub>CBREQ</sub> [15:0], DMA <sub>CSREQ</sub> [15:0], DMA <sub>CLBREQ</sub> [15:0], та DMA <sub>CLSREQ</sub> [15:0] сигналів –запитів. Встановлений біт вимикає логіку синхронізації для заданої групи DMA–запитів	0x0000
1:16	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів невизначений.	N/A

### 3.9.8.2.1 Канальні регістри адреси джерела (Channel Source Address Registers, DMA<sub>CC0SrcAddr</sub> – 0xFFE0 4100 та DMA<sub>CC1SrcAddr</sub> – 0xFFE0 4120)

Два регістри DMA<sub>CCxSrcAddr</sub> (таблиця 3.26) доступні для читання/запису зберігають поточну адресу джерела даних для передачі, яку вирівняно побайтово. Кожен з регістрів програмується напряму за допомогою програми перед тим, як увімкнути канал. Коли DMA – канал увімкнений, цей регістр оновлюється:

- при інкременті адреси джерела,
- при переході за елементами зв’язаного списку, коли завершується передача даних.

Коли канал активний, читання регістра не дасть корисної інформації. Значення може змінитися за завершенням читання регістра програмою. Регістр призначений для читання тільки тоді, коли канал неактивний: значення регістра вказуватиме на адресу останнього прочитаного елемента.

Примітка: Адреси джерела та приймача мають бути вирівняно за розміром джерела та приймача.

Таблиця 3.26 – Опис розрядів регістрів DMA<sub>CCxSrcAddr</sub>

Біт	Символ	Опис	Початкове значення
31:0	SrcAddr	Адреса джерела DMA	0x0000 0000

### 3.9.8.2.2 Канальні регістри адреси приймача (Channel Destination Address Registers, DMA<sub>CC0DestAddr</sub> – 0xFFE0 4104 та DMA<sub>CC1DestAddr</sub> – 0xFFE0 4124)

Два регістри DMA<sub>CCxDestAddr</sub> (таблиця 3.27) доступні для читання/запису і зберігають поточну адресу приймача даних, яку вирівняно побайтово. Кожен з регістрів програмується напряму за допомогою програми перед тим, як увімкнути канал. Коли DMA – канал увімкнений, регістр оновлюється з інкрементом адреси приймача та при переході за елементами зв'язаного списку, коли завершується передача даних. Коли канал активний, читання регістра не дасть корисної інформації. Значення може змінитися за завершенням читання регістра програмою. Регістр призначений для читання тільки тоді, коли канал неактивний: значення регістра вказуватиме на адресу приймача останнього прочитаного елемента.

Таблиця 3.27 – Опис розрядів регістрів DMA<sub>CCxDestAddr</sub>

Біт	Символ	Опис	Початкове значення
31:0	DestAddr	Адреса приймача DMA	0x0000 0000

### 3.9.8.2.3 Канальні регістри елемента зв'язаного списку (Channel Linked List Item Registers, DMA<sub>CC0LLI</sub> – 0xFFE0 4108 та DMA<sub>CC1LLI</sub> – 0xFFE0 4128)

Два регістри DMA<sub>CCxLLI</sub> (таблиця 3.28) доступні для читання/запису і зберігають вирівняну за словом адресу наступного елемента зв'язаного списку (Linked List Item, LLI). Якщо LLI дорівнює 0, тоді поточний LLI є останнім у ланцюгу, і DMA – канал відключається за завершенням всіх пов'язаних зі зв'язаним списком DMA – передач.

Примітка: Програмування даного регістра тоді, коли увімкнений DMA – канал, може призвести до невизначених наслідків.

Таблиця 3.28 – Опис розрядів регістрів DMASSxLLI

Біт	Символ	Опис	Початкове значення
0	Reserved	Зарезервований, зчитувати як 0, не змінювати	N/A
1	R	Зарезервований, записувати як 0, маскований при читанні	0
31:2	LLI	Елемент зв'язаного списку. Біти [31:2] адреси вказують на наступний LLI. Біти [1:0] адреси 0	0

#### 3.9.8.2.4 Регістри керування каналом (Channel Control Registers, DMASS0Control – 0xFFE0 410C та DMASS0Control – 0xFFE0 412C)

Два регістри DMASSxControl (таблиця 3.29) доступні для читання/запису і зберігають керуючу інформацію DMA – каналу, таку як розмір передачі, розмір пакету, ширина передачі.

Таблиця 3.29 – Опис розрядів регістрів DMASSxControl

Біт	Символ	Опис	Початкове значення
11:0	TransferSize	Розмір передачі. Запис встановлює розмір передачі у випадку, коли DMA виступає в ролі контролера потоку. Читання даного поля повертає кількість завершених передач у шині приймача. Поле призначене для читання, коли канал відключений.	0
14:12	SBSize	Розмір пакету джерела. Відображає кількість передач, які разом складають запит на пакетну передачу джерела.	0
17:15	DBsize	Розмір пакету приймача. Відображає кількість передач, які разом складають запит на пакетну передачу приймача. Значення має відповідати розміру пакету периферійного пристрою приймача, або, якщо приймачем є пам'ять, граничному розміру пам'яті.	0
20:18	SWidth	Ширина передачі джерела. Передачі ширше за розмір шини АНВ – майстра не підтримуються. Ширина джерела та приймача можуть відрізнятися одне від одного. Обладнання автоматично пакує та розпаковує дані так, як необхідно.	0



Продовження таблиці 3.29

Біт	Символ	Опис	Початкове значення
23:21	DWidth	Ширина передачі приймача. Передачі ширше за розмір шини АНВ – майстра не підтримуються. Ширина джерела та приймача можуть відрізнятися одне від одного. Обладнання автоматично пакує та розпаковує дані так, як необхідно	0
25:24	–	Зарезервовані, запис до бітів неможливий. Результат читання бітів невизначений.	N/A
26	SI	Інкремент джерела. Якщо встановлений, адреса джерела буде інкрементована після кожної передачі.	0
27	DI	Інкремент приймача. Якщо встановлений, адреса приймача буде інкрементована після кожної передачі.	0
30:28	Prot	Біти захисту та доступу	0
31	TCIE	Біт увімкнення переривання термінального рахунку. Контролює, чи повинний поточний LLI викликати переривання термінального рахунку.	0

Кожен з регістрів програмується напряму за допомогою програми перед тим, як увімкнути канал. Якщо канал увімкнений, регістр оновлюється при переході до наступного елементу зв'язаного списку за завершенням передачі повного пакету даних. Коли канал активний, читання регістра не дасть корисної інформації. Значення може змінитися за завершенням читання регістра програмою. Регістр призначений для читання тільки тоді, коли канал неактивний.

Вплив значення 3 – бітних полів DBSize та SBSIZE на відповідні розміри пакетів наведено в таблиці 3.30.

Таблиця 3.30 – Вплив значення 3 – бітних полів DBSize та SBSIZE на відповідні розміри пакетів

Значення в бітах	Розмір пакету
000	1
001	4
010	8
011	16
100	32
101	64
110	128
111	256

Вплив значення 3 – бітних полів SWidth та DWidth на відповідну ширину передач наведено в таблиці 3.31.

Таблиця 3.31 – Вплив значення 3 – бітних полів SWidth та DWidth на відповідну ширину передач

Значення в бітах	Ширина передачі
000	Байт (8 біт)
001	Півслова (16 біт)
010	Слово (32 біт)
011 або 1xx	Зарезервовано

#### 3.9.8.2.5 Захист та доступ до інформації

АНВ надає доступ до інформації джерелу та приймачу тоді, коли проходить передача. Інформація про передачу надається при програмуванні DMA – каналу (біт Prot регістра DMACCxControl, а також біт Lock регістра DMACCxConfiguration) (таблиця 3.32). Ці біти програмуються у програмі, інформація може бути використана в разі необхідності.

Таблиця 3.32 – Опис бітів захисту та доступу

Біт DMASSxControl	Значення	Опис	Початкове значення
28		Привілейований або користувацький режим. Даний біт контролює сигнал АНВ HPROT[1]. Відображає режим доступу	0
	0	Користувацький режим	
	1	Привілейований режим	
29		Відображає, чи є доступ буферизованим. Може бути використаний - для повідомлення АМВА – мосту, що читання може бути завершено без очікувань. Контролює сигнал АНВ HPROT[2].	0
	0	Не буферизований	
	1	Буферизований	
30		Відображає, чи є доступ кешованим. Контролює сигнал АНВ HPROT[3]. Може бути використаний для повідомлення АМВА – мосту, що якщо зчитано перший пакет розміру 8, міст може передати повний пакет розміру 8 на шину приймача, аніж передаватиме транзакції одну за раз.	0
	0	Не кешований	
	1	Кешований	

### 3.9.8.2.6 конфігурації каналу (Channel Configuration Registers, DMASS0Configuration – 0xFFE0 4110 та DMASS1Configuration – 0xFFE0 4130)

Два регістри DMASSxConfiguration (таблиця 3.33) доступні для читання/запису, за виключенням 17 біта, що доступний тільки для читання. Використовується для конфігурації DMA – каналу. Регістри не оновлюються при запиті нового LLI.

Таблиця 3.33 – Опис розрядів регістрів DMACCxConfiguration

Біт	Символ	Значення	Опис	Початкове значення
0	E (Enable)		Стан каналу також може бути зчитаний у регістрі DMACEnbldChns. Канал вмикається встановленням даного біта. Вимикається канал скиданням біта. Це призведе до закінчення поточної передачі та втрати даних у FIFO відповідного каналу. Перезапуск каналу встановленням біта Enable може призвести до неочікуваних результатів. Канал також вимикається тоді, коли досягнуто останній елемент LLI або якщо виникла помилка.	0
		0	Відключений	
		1	Увімкнений	
4:1	SrcPeripheral		Джерело запитів периферії	0
		0000	SSP0 Tx	
		0001	SSP0 Rx	
		0010	SSP1 Tx	
		0011	SSP1 Rx	
		0100	SD/MMC	
		0101	I2S channel 0	
		0110	I2S channel 1	
		0111		
		або 1xxx	Зарезервовані значення	
5	–	–	Зарезервовані значення.	N/A

Продовження таблиці 3.33

Біт	Символ	Значення	Опис	Початкове значення
9:6	DestPeripheral		Ці біти визначають приймач DMA – запитів. Поле ігнорується у випадку, якщо приймачем передачі є пам'ять. Див. опис значень SrcPeripheral.	0
10	–	–	Зарезервовані значення.	N/A
13:11	FlowCntrl		Контроль потоку та тип передачі. В якості контролера потоку можуть виступати: DMA, джерело, приймач. Типи передач наведено в таблиці 3.34.	0
14	IE (Interrupt Error)		Маска переривання «помилки». Якщо біт скинутий, він маскує переривання «помилки» відповідно го каналу.	0
15	ITC		Маска переривання термінального рахунку. Якщо біт скинутий, він маскує переривання термінального рахунку відповідного каналу.	0
16	L (Lock)		Блокування. Якщо біт встановлений, він дозволяє захищені передачі.	0
17	A (Active)		Біт може бути використаний для коректного завершення роботи каналу DMA. Запис до біта нічого не змінює.	
		0	У FIFO черзі каналу даних немає	
		1	У FIFO черзі каналу є дані	
18	H (Halt)			
		0	Увімкнути DMA – запити.	
		1	Ігнорувати DMA – запити джерела.	
31:19	–		Зарезервовані значення.	

Вплив 3 – бітних значень поля FlowCntrl на вибір контролера потоку та типу передачі наведено в таблиці 3.34.

Таблиця 3.34 – Вплив значень поля FlowCtrl на вибір контролера потоку та типу передачі

Значення в бітах	Тип передачі	Контролер
000	Пам'ять – пам'ять	DMA
001	Пам'ять – периферійний пристрій	DMA
010	Периферійний пристрій – пам'ять	DMA
011	Периферійний пристрій джерела – периферійний пристрій приймача	DMA
100	Периферійний пристрій джерела – периферійний пристрій приймача	Периферійний пристрій приймача
101	Пам'ять – периферійний пристрій	Периферійний пристрій
110	Периферійний пристрій – пам'ять	Периферійний пристрій
111	Периферійний пристрій джерела – периферійний пристрій приймача	Периферійний пристрій джерела

## ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- 1) Опишіть призначення модуля прямого доступу до пам'яті в мікропроцесорних системах.
- 2) Як контролер DMA підключається до окремих вузлів мікроконтролера?
- 3) Скільки модулів DMA має контролер DMA?
- 4) Як контролер DMA підключається до шини АНВ?
- 5) Які види пересилань даних виконуються контролером DMA?
- 6) Як в контролері DMA відбувається пересилання даних з пам'яті в пам'ять?
- 7) На які групи розбито регістри керування контролером DMA?
- 8) Як здійснюється дозвіл роботи контролера DMA?
- 9) Які переривання має кожний з модулів DMA?
- 10) Які регістри призначені для програмування переривань від модулів
- 11) Як виконується синхронізація модулів DMA?
- 12) Як програмується пересилання даних з пам'яті в пам'ять?
- 13) Як здійснюється пересилання даних в пакетному режимі?
- 14) Як програмується використання периферійних пристроїв в якості контролера потоку в модулях DMA?
- 15) Як в модулях DMA підтримується пересилання несуміжних даних?
- 16) Назвіть регістри, які призначено для керування модулем DMA.
- 17) Опишіть призначення окремих розрядів регістра статусу переривань.
- 18) Опишіть призначення окремих розрядів регістрів статусу, скидання та статусу необробленого переривання термінального рахунку.
- 19) Опишіть призначення окремих розрядів регістрів статусу, скидання та статусу необробленого переривання за помилкою.
- 20) Опишіть призначення окремих розрядів регістрів запиту програми на окремі види передачі.
- 21) Опишіть призначення окремих розрядів регістра увімкнення каналу.
- 22) Опишіть призначення окремих розрядів каналних регістрів.