

Лекція 2 ОДНОКРИСТАЛЬНІ МІКРОКОНТРОЛЕРИ AVR З RISC-АРХІТЕКТУРОЮ

Регістр статусу містить інформацію про результати виконання останньої арифметичної або логічної команди та є аналогом регістру прапорців у мікроконтролерах інших серій. Ця інформація може використовуватися для того, щоб змінити процес виконання програми в командах умовного передавання керування та інших умовних операторах.

Регістр статусу SREG має наступний формат:

Номер біту	7	6	5	4	3	2	1	0
Позначення біту	I	T	H	S	V	N	Z	C

I - загальний дозвіл переривань. При встановленні цього прапорця в одиничний стан дозволяється робота всієї системи переривань. Якщо прапорець скинутий (тобто встановлений в нуль) всі переривання заборонені. Прапорець I апаратно скидається відразу після виклику відповідної процедури обробки переривання й встановлюється при виконанні команди RETI, дозволяючи наступні переривання. Прапорець I може бути також встановлений і скинутий програмно за допомогою команд SEI і CLI відповідно.

T – біт користувача для тимчасового зберігання інформації. Біт використовується командами BLD (завантаження біта T) і BST (читання біта T) як елемент пам'яті для тимчасового зберігання інформації. Будь-який біт регістра загального призначення може бути скопійований у T, а потім вміст T може бути скопійований в будь-який інший біт того ж або іншого регістра.

H - прапорець половинного перенесення. Цей прапорець встановлюється в одиницю, якщо мало місце перенесення із молодшої

половини байта (з 3-го розряду в 4-й) або позика зі старшої половини байта при виконанні деяких арифметичних операцій.

S - прапорець знаку. Цей прапорець є результатом операції «ВИКЛЮЧАЛЬНЕ АБО» (XOR) між прапорцями **N** (від'ємний результат) і **V** (переповнення числа в додатковому коді). Відповідно, цей прапор встановлюється в одиницю, якщо результат виконання арифметичної операції менше нуля.

V - прапорець переповнення додаткового коду. Цей прапорець використовується при роботі зі знаковими числами (числами, які представлені у додатковому коді). Прапорець встановлюється в одиницю, якщо в результаті арифметичної операції відбувається переповнення числа, представленого у додатковому коді.

N - прапорець від'ємного значення. Цей прапорець встановлюється в одиницю, якщо в результаті арифметичної операції старший розряд результату дорівнює одиниці. Якщо старший розряд результату дорівнює нулю, то прапорець **N** теж дорівнює нулю.

Z - прапорець нуля. Цей прапорець встановлюється в одиницю, якщо результат операції, яка виконана, дорівнює нулю.

C - прапорець перенесення. Цей прапорець встановлюється в одиницю, якщо має місце переповнення результату (перенесення у старший розряд) при виконанні арифметичної операції.

Вплив різних команд на прапорці регістра статусу можна визначити за допомогою таблиці 2.4, у якій наведена система команд мікроконтролерів AVR.

При переході на підпрограму обробки переривання регістр статусу автоматично не зберігається. Програміст повинен передбачити у підпрограмі обробки переривання команди, які будуть зберігати вміст цього регістра на початку та відновлювати його в кінці підпрограми.

3. Пам'ять мікроконтролерів AVR

Мікроконтролери AVR мають в своєму складі три види пам'яті (рис. 2.1). По-перше, це оперативний запам'ятовувальний пристрій ОЗП (пам'ять даних). У документації фірми *Atmel* ця пам'ять називається SRAM. Об'єм ОЗП для різних мікроконтролерів змінюється від повної її відсутності до 2 Кбайт (див. табл.2.1).

Другий вид пам'яті - це постійний запам'ятовувальний пристрій ПЗП (пам'ять програм). Вона виконана за Flash-технологією і призначена для зберігання керуючої програми. Об'єм програмної пам'яті в різних мікросхемах складає від 1 до 64 Кбайт (табл. 2.1). Пам'ять програм допускає стирання записаної інформації і повторний запис. Проте кількість циклів запису/стирання обмежена (до 1000 циклів).

Третій вид пам'яті — це енергонезалежна пам'ять даних. Вона також виконана за Flash-технологією, але в технічній документації вона називається EEPROM. Основне призначення цього виду пам'яті - довготривале зберігання даних навіть при вимкненому джерелі живлення. Пам'ять EEPROM допускає до 100000 циклів запису/стирання, а кількість циклів читання не обмежена. Об'єм пам'яті EEPROM порівняно невеликий і складає від 64 байт до 4 Кбайт (табл. 1).

Кожен з цих трьох видів пам'яті має свій власний адресний простір, і доступ до різних видів пам'яті здійснюється незалежно один від одного. Така побудова мікроконтролерів називається архітектурою Гарвардського типу.

Пам'ять програм призначена для зберігання керуючої програми мікроконтролера. Кожен елемент цієї зберігає одне 16-розрядне двійкове число. Кожне таке число містить як код операції, так і один або декілька параметрів команди. Крім того, в пам'яті програм, зазвичай, зберігають деякі постійні константи, таблиці символів та інші незмінні величини.

Для різних МК пам'ять програм має різний об'єм (табл. 2.1), проте при будь-якому об'ємі пам'яті вона є безперервною областю і починається з комірки, що має нульову адресу.

Деякі адреси програмної пам'яті зарезервовані, тобто використовуються для спеціальних цілей. Першою зарезервованою адресою можна вважати нульову адресу. Саме з цієї адреси починається виконання програми після системного скидання мікроконтролера. Решта зарезервованих адрес - це вектори переривань.

Вектор переривання - це адреса в програмній пам'яті, з якої починається виконання процедури (підпрограми) обробки переривання. Оскільки будь-який МК AVR має декілька джерел переривань (табл. 1), то і векторів переривання теж декілька. Докладніше система переривань МК AVR розглядається нижче.

Пам'ять даних МК AVR (оперативний запам'ятовувальний пристрій) займає окремий адресний простір з адресами від \$0000 до \$FFFF. Тобто максимальний об'єм пам'яті, що адресується, дорівнює 64 Кбайт. Проте більшість мікроконтролерів мають набагато меншу пам'ять. Структура ж пам'яті завжди однакова. Оперативна пам'ять мікроконтролерів AVR поділяється на три області:

\$0000-\$001F - область пам'яті, суміщена з регістрами загального призначення РЗП (рис. 2);

\$0020-\$005F - область пам'яті, суміщена з регістрами введення-виведення (PBB);

\$0060-\$FFFF - не суміщена ні з чим область пам'яті.

Останню область можна розділити на *область внутрішнього ОЗП* (адреси \$0060 - RAMEND) і *область зовнішнього ОЗП* (адреси RAMEND+1 - \$FFFF). Під RAMEND розуміється адреса останньої комірки внутрішнього

ОЗП конкретного мікроконтролера. Об'єм внутрішнього ОЗП для різних МК різний і саме він наведений у стовпці «SRAM» табл. 1.

Більшість МК AVR мають лише внутрішній вбудований ОЗП. Проте до складу серії AVR входять МК, що допускають підключення зовнішніх мікросхем ОЗП. В результаті об'єм ОЗП МК може бути розширений до 64 Кбайт.

Область пам'яті з адреси \$0020 по адресу \$005F суміщена з регістрами введення-виведення. Ці регістри дозволяють центральному процесору обмінюватися інформацією з вбудованими периферійними пристроями самого МК, такими, як таймери, компаратори, канали послідовної передачі інформації, система переривань, АЦП і т.п. Максимально можлива кількість РВВ рівна 64. Реальна кількість РВВ майже завжди значно менше їх максимально можливої кількості. Проте дана область пам'яті завжди використовується тільки для цієї мети. Якщо регістр існує, то існує і відповідна комірка пам'яті. Решта комірок цієї області ОЗП просто відсутня.

Енергонезалежна пам'ять даних (EEPROM) - це спеціальна внутрішня пам'ять, яка виконана за Flash-технологією і призначена для довготривалого зберігання даних. У сучасних мікроконтролерних пристроях часто виникає необхідність у зберіганні таких даних. Для подібних завдань зазвичай не вимагається великих об'ємів пам'яті, тому МК AVR мають об'єм EEPROM від 64 байт до 4 Кбайт (табл. 1).

До цієї пам'яті центральний процесор МК звертається не так, як до решти видів пам'яті. Для ЦП не існує адресного простору EEPROM і до неї він звертається за допомогою регістрів введення-виведення. Для МК з об'ємом EEPROM менше 256 байт таких регістрів три:

EEAR — регістр адреси EEPROM. Працює тільки на запис. За допомогою цього регістра МК вибирає комірку, куди потрібно записати або звідки потрібно прочитати дані;

EEDR — реєстр даних EEPROM. Працює як на запис, так і на читання. Через цей реєстр в EEPROM поступає байт, який записується. Через нього ж процесор отримує байт при читанні з EEPROM;

EECR — реєстр керування, який визначає режими роботи EEPROM. Саме через нього подаються команди читання і запису EEPROM.

Якщо об'єм EEPROM перевищує 256 байт, то замість одного реєстра адреси (EEAR) такий МК має два реєстри: EEARH і EEARL. Реєстри доступу до EEPROM мають наступні номери:

EEAR - \$1E; EEARL - \$1E; EEARH - \$1F; EEDR - \$1D; EECR - \$1C.

Способи програмування Flash- і EEPROM- пам'яті

Мікроконтролери AVR допускають декілька способів програмування Flash- і EEPROM- пам'яті. Основні способи такі:

- паралельне програмування (Self-Prog);
- послідовне програмування з використанням SPI-інтерфейса.

При паралельному програмуванні програматор передає в мікросхему записувані дані побайтно за допомогою 8-розрядної шини.

При послідовному програмуванні використовується спеціальний послідовний інтерфейс, що отримав назву SPI. За допомогою цього інтерфейсу дані передаються в мікросхему послідовно, біт за бітом, з використанням всього трьох ліній. Послідовний спосіб набагато повільніший, ніж паралельний. Але він більш універсальний і допускає програмування мікросхеми без виймання ВІС з плати. У табл. 1 у стовбці «ISP (I), Self-Prog (S)» для кожної мікросхеми показані підтримувані способи програмування.

Записувати інформацію в EEPROM можна також за допомогою програматора. Причому для запису інформації в пам'ять програм і в EEPROM використовується один і той же програматор. Такий порядок доступу до

пам'яті дозволяє при необхідності відмовитися від програмного перезапису EEPROM і використовувати цю пам'ять для зберігання будь-яких незмінних констант, що збільшує гнучкість системи.

4 Порти введення-виведення

Мікроконтролери AVR мають в своєму складі від одного до семи паралельних портів введення-виведення, які призначені для обміну даними із зовнішніми пристроями. Кожен розряд такого порту приєднаний до одного з виводів (контактів) МК. Порти можуть бути повні і неповні. Повний порт містить 8 розрядів. У неповних портах можуть бути задіяні сім, шість або навіть три розряди. Але для центрального процесора порти залишаються 8-розрядними, бо він завжди записує в такі порти і читає з них повноцінний байт даних. Розряди, які не використовуються, при записуванні просто втрачаються, а при читанні їх значення дорівнює нулю. Кожен порт має своє ім'я, яким є латинська буква від А до G (рис. 1).

Для керування кожним портом введення-виведення використовується три спеціальних РВВ. Це регістри PORTx, DDRx і PINx. Під «x» тут мається на увазі конкретна буква - ім'я порту. Наприклад, для порту А імена регістрів управління будуть такими: PORTA, DDRA і PINA.

Розглянемо призначення кожного з цих регістрів:

PORTx — регістр даних (використовується для виведення інформації);

DDRx — регістр напряму передачі інформації;

PINx — регістр введення інформації.

Окремі розряди цих регістрів також мають свої імена. Розряди регістра PORTx зазвичай іменуються як Pxn, де «n» - це номер розряду. Наприклад, розряди регістра PORTA іменуватимуться таким чином: PA0,

PA1, PA2—PA7. Розряди регістру DDRx іменуються як DDxp, а розряди регістру PINx іменуються як PINxp.

Будь-який порт введення-виведення МК AVR влаштований таким чином, що кожен його розряд може працювати як на введення, так і на виведення. Для перемикання напряму передачі служить регістр DDRx. Кожен розряд регістра DDRx керує своїм розрядом порту. Якщо в якому-небудь розряді регістра DDRx записаний нуль, то відповідний розряд порту працює як вхід. Якщо ж в цьому розряді одиниця, то розряд порту працює як вихід.

Для того, щоб видати інформацію на зовнішній контакт МК, потрібно у відповідний розряд DDRx записати логічну одиницю, а потім записати байт даних в регістр PORTx.

Для того, щоб прочитати інформацію із зовнішнього контакту МК, потрібно спочатку перевести потрібний розряд порту в режим введення, тобто записати у відповідний розряд регістра DDRx нуль. Тільки після цього на даний контакт МК можна подавати цифровий сигнал від зовнішнього пристрою. Далі МК читає байт з регістра PINx.

Порти введення-виведення МК AVR мають ще одну корисну функцію. У режимі введення інформації вони можуть при необхідності підключати до кожного виводу порту внутрішній резистор навантаження. Внутрішній резистор дозволяє значно розширити можливості порту. Такий резистор створює для зовнішніх пристроїв струм, що витікає. Завдяки цьому резистору спрощується підключення зовнішніх контактів і кнопок. Включенням і відключенням внутрішніх резисторів керує регістр PORTx, якщо порт знаходиться в режимі введення. Це видно з табл. 2, в якій показані всі режими роботи порту.

Конфігурація порту введення-виведення Таблиця 2.

DDxn	Rxn	Режим	Резистор	Примітка
0	0	Вхід	Відключений	Вивід відключено від схеми
0	1	Вхід	Підключений	Вивід є джерелом струму
1	0	Вихід	Відключений	На виході «0»
1	1	Вихід	Відключений	На виході «1»

На рис. 3. показана спрощена схема одного розряду порту введення-виведення. Ця схема є універсальною частиною схеми виводу порту та дає уявлення про роботу порту.

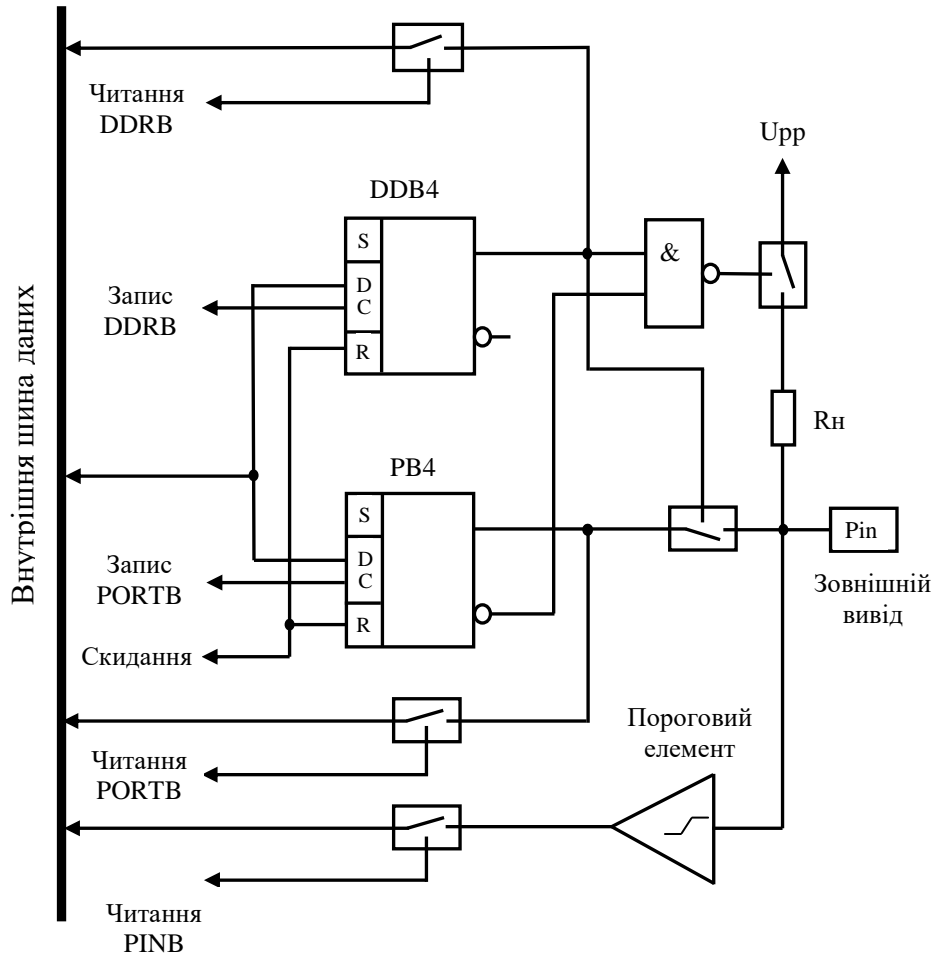


Рис. .3. Схема розряду порту введення-виведення

Послідовні порти введення-виведення

Деякі мікроконтролери AVR мають:

- вбудований універсальний послідовний асинхронний приймач-передавач (UART);
- універсальний послідовний синхронно/асинхронний приймач-передавач (USART).

Наявність таких портів для різних МК вказана в табл. 1. Канали UART (USART) призначені для обміну інформацією між МК і будь-яким зовнішнім пристроєм. Протокол UART (USART) - це досить поширений протокол послідовної передачі інформації. Такий протокол, зокрема, використовує послідовний порт комп'ютера. За допомогою UART (USART) можна організувати лінію зв'язку не тільки між двома МК, але і між МК та комп'ютером.

Для обміну інформацією UART (USART) використовує дві лінії: RxD і TxD. Одна лінія використовується для приймання інформації, інша - для передавання. В модулях UART посилка може бути 8- або 9-розрядною. В модулі USART її довжина може складати від 5 до 9 розрядів. Крім того, модулі можуть виробляти і контролювати розряд парності.

Швидкість передавання визначається спеціальним внутрішнім програмованим подільником і частотою тактового генератора МК. Коефіцієнт ділення подільника може змінюватися від 2 до 65536. Для обміну інформацією через послідовний порт необхідно так підібрати коефіцієнт ділення і частоту тактового генератора, щоб отримати одну із стандартних швидкостей передачі інформації (2400, 4800, 9600, 14400, 19200, 28800 біт в секунду).

Послідовний периферійний інтерфейс (SPI) - це спеціальний послідовний інтерфейс, розроблений для зв'язку мікроконтролерів між собою. Канал SPI використовує для передачі інформації три лінії:

- лінію MISO (Master Input / Slave Output);
- лінію MOSI (Master Output / Slave Input);
- лінію SCK (Тактовий сигнал).

В мікроконтролерах AVR канал SPI може виконувати дві функції. По-перше, за допомогою цього інтерфейсу можна не тільки організувати послідовний канал обміну інформацією між двома МК, але і між МК і будь-яким периферійним пристроєм, що має SPI-інтерфейс. Існує цілий набір подібних пристроїв: цифрові потенціометри, ЦАП/АЦП, зовнішня Flash-пам'ять та ін.

Друге призначення каналу SPI - програмування мікроконтролера. Саме через цей канал здійснюється послідовне програмування пам'яті програм і пам'яті EEPROM. Перевага програмування через SPI полягає в тому, що такий спосіб дозволяє програмувати МК, не виймаючи його з пристрою. Це так зване внутрішньосхемне програмування. Зазвичай на платі пристрою, який налагоджується, передбачають спеціальні контакти, куди і підключається програматор.

Послідовний дводротовий інтерфейс (TWI) є повним аналогом шини I²S фірми *Philips*, що отримала широке розповсюдження в різних системах керування побутовою і промисловою технікою. Інтерфейс дозволяє об'єднати разом до 128 пристроїв, підключивши їх до однієї дводротової шини. Шина I²S містить лінію тактового сигналу SCL та лінію передавання даних SDA.

Інтерфейс дозволяє обмінюватися даними між ведучим пристроєм, яким звичайно є МК, і будь-яким із зовнішніх пристроїв, підключених до дводротової лінії. При цьому ведучий пристрій може як передавати дані на відомий пристрій, так і приймати дані з нього.

Наявність інтерфейсу для роботи з I²C шиною дозволяє застосовувати МК в системах керування телевізорів, радіоприймачів, магнітол, рідкокристалічних дисплеїв на і т.п.

5. Система переривань

Система переривань мікроконтролерів AVR призначена для обслуговування декількох джерел переривань, які поділяються на внутрішні і зовнішні.

Внутрішнім перериванням називається переривання, викликане одним з вбудованих периферійних пристроїв самого МК (наприклад, переривання по таймеру, аналоговому компаратору, АЦП і так далі).

Зовнішнім перериванням називається переривання, викликане сигналом, що поступає від зовнішнього джерела на спеціальний вхід МК. Кількість внутрішніх та зовнішніх переривань для різних МК AVR наведена у табл. 1.

Керування системою переривань здійснюється за допомогою спеціальних регістрів введення-виведення. Визначальним регістром тут є регістр статусу SREG, який розглядався вище. Сьомий біт цього регістру (біт I) є прапорцем загального дозволу переривань. Коли цей прапорець скинутий (тобто містить логічний нуль), всі переривання у МК заборонені. Для дозволу переривань потрібно встановити цей прапорець в одиницю.

Проте частіше всі види переривань не потрібні одночасно. Для того, щоб заборонити одні переривання і дозволити інші, застосовуються так звані маскуючі регістри (регістри маски). Регістр маски - це звичайний регістр введення-виведення, який призначений для керування окремими джерелами переривань. Кожному біту в регістрі маски відповідає одне джерело. Якщо біт скинуто в нуль, переривання цього виду заборонено. Якщо біт встановлений в одиничний стан, переривання дозволено.

В МК AVR застосовуються два регістри маски. Регістр GIMSK керує всіма видами переривань, окрім переривань від таймерів. В деяких МК сімейства «Мega» цей регістр називається GICR. Для керування перериваннями від таймерів є спеціальний регістр TIMSK.

Окрім регістрів маски для керування процесом виконання переривань існують ще два регістри. Це регістри прапорців переривань. Кожний біт такого регістра - це прапорець одного з видів переривань. Під час надходження запиту на переривання прапорець встановлюється в одиницю. За станом прапорця програма може визначити наявність запиту.

В певних режимах після встановлення прапорця процедура (підпрограма) обробки переривання викликається автоматично. Відразу після виклику процедури відповідний прапорець скидається. Мікроконтролери AVR мають два регістри прапорців: регістр GIFR (обслуговує ті ж переривання, що і регістр GIMSK) і регістр TIFR (прапорці переривань від таймерів).

Алгоритм роботи системи переривань

Загальний алгоритм роботи системи переривань наступний. Після скидання мікроконтролера всі переривання заборонені (прапорці дозволу скинуті). Якщо програміст планує використовувати один з видів переривань, він повинен в своїй програмі встановити прапорець I загального дозволу переривань регістра статусу SREG в одиницю і дозволити це переривання шляхом встановлення у одиницю відповідного прапорця дозволу у регістрі маски.

Під час надходження запиту на переривання встановлюється прапорець відповідного переривання. Прапорець встановлюється навіть в тому випадку, якщо переривання заборонено. Якщо переривання дозволено, то МК приступає до його виконання. Поточна програма

тимчасово припиняється, і керування передається за адресою відповідного вектора переривання на підпрограму обробки переривання.

В той же момент прапорець I автоматично скидається, забороняючи обробку інших переривань. Прапорець, який відповідає викликаному перериванню, також скидається, сигналізуючи про те, що мікроконтролер вже приступив до його обробки. Підпрограми обробки переривання обов'язково повинні закінчуватися командою повернення з переривання (RETI). По цій команді керування передається в ту точку основної програми, у якій перервалася її робота. Прапор I при цьому автоматично встановлюється в одиницю, дозволяючи нові переривання.

Слід відмітити, що без вживання спеціальних заходів неможливі вкладені переривання. Поки обробляється одне переривання, вся решта переривань заборонена. Проте жодне переривання не залишається без обробки. При отриманні запиту на переривання відповідний прапорець обов'язково встановлений. В цьому стані він знаходитиметься до тих пір, поки дане переривання не буде оброблено.

Після закінчення обробки чергового переривання відбувається перевірка решти прапорців, і якщо є хоч одне необроблене переривання, МК переходить до його обробки. Якщо необроблених переривань виявиться декілька, то застосовується закон пріоритетів. Зі всіх переривань вибирається те переривання, пріоритет якого вище. Чим менше адреса вектора переривання, тим вище його пріоритет.

У табл. 3 як приклад наведені адреси векторів переривань для МК сімейства «Tiny».

**Адреси векторів переривань
мікроконтролерів сімейства «Tiny»**

Таблиця 3.

Джерело	Опис	Tiny11x	Tiny12x	Tiny15L	Tiny28x
INT0	Зовнішнє переривання 0	\$001	\$001	\$001	\$001
INT1	Зовнішнє переривання 1	-	-	-	\$002
PIN_CANGE	По зміні сигналу на будь-якому з виводів	\$002	\$002	\$002	-
LOW_LEVEL	По низькому рівню на вході порту В	-	-	-	\$003
TIMER1 COMPА	По співпаданню вмісту таймера-лічильника Т1 з вмістом контрольного регістра	-	-	\$003	-
TIMER1 OVF	Переповнення таймера-лічильника Т1	-	-	\$004	-
TIMER0 OVF	Переповнення таймера-лічильника Т0	\$003	\$003	\$005	\$004
EE_RDY	По готовності EEPROM	-	\$004	\$006	-
ANA_COMP	По сигналу від аналогового компаратора	\$004	\$005	\$007	\$005
ADC	По завершенню перетворення в АЦП	-	-	\$008	-