

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

**МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ
ЛАБОРАТОРНИХ РОБІТ**

з курсу „Мікропроцесорна техніка”

Мікроконтролери сімейства STMicroelectronics

Київ 2008

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

МЕТОДИЧНІ ВКАЗІВКИ
з курсу „Мікропроцесорна техніка”
Архітектура мікроконтролерів
STMicroelectronics

ДЛЯ СТУДЕНТІВ СПЕЦІАЛЬНОСТІ
7.09.08.03 – "ЕЛЕКТРОННІ СИСТЕМИ"
УСІХ ФОРМ НАВЧАННЯ

Затверджено Радою факультету
електроніки, протокол № 02/08
від 25.02.2008 р.

Київ НТУУ “КПІ” 2008

Методичні вказівки з курсу „Мікропроцесорна техніка”
„Архітектура мікроконтролерів STMicroelectronics” для студентів спеціальності 7.09.08.03 –
"Електронні системи" всіх форм навчання. - К.: НТУУ “КПІ”, 2008. – 40 с.

Навчальне видання

Методичні вказівки з курсу „Мікропроцесорна техніка”
„Архітектура мікроконтролерів STMicroelectronics”
для студентів спеціальності 7.09.08.03 – "Електронні системи" всіх форм навчання

Укладач

Терещенко Тетяна Олександрівна
Петергеря Юлія Сергійовна
Хохлов Юрій Віталіович

Рецензенти:

В.В. Рогаль, доц., канд.техн.наук
В.А. Тодоренко, доц., канд.техн.наук

Редактор

ЗМІСТ

ВСТУП	5
Загальні відомості про мікроконтролери сімейства ST7	6
Переривання	17
Режими енергоспоживання	20
Порти введення-виведення	23
Послідовний периферійний інтерфейс SPI	25
Аналого-цифровий перетворювач	28
Таймери	30
Сторожовий таймер (WDG)	30
Таймер ART	31

ВСТУП

Метою вивчення мікроконтролерів ST7 є розширення доступного інструментарія майбутніх інженерів та науковців у сфері мікропроцесорної техніки. У методичних вказівках викладені теоретичні відомості про основні принципи побудови і функціонування мікропроцесорних пристроїв на базі однокристальних мікроконтролерів сімейства ST7.

Дані методичні вказівки містять інформацію про ядро, пам'ять та периферію мікроконтролерів, а саме портам введення та виведення інформації, системи переривань, інтерфейсу SPI, таймеру та АЦП. Для закріплення матеріалу наводяться приклади.

Матеріали даних методичних вказівок може бути використано і при дипломному проектуванні.

Загальні відомості про мікроконтролери сімейства ST7

Однокристалні мікроконтролери сімейства ST7 є 8-бітними RISC-контролерами загального призначення з ядром промислового стандарту, розширеного компанією STMicroelectronics у порівнянні з AVR мікроконтролерами шляхом залучення додаткових команд та удосконалення модуля переривань. Система команд має 63 команди з 17 способами адресації з можливістю 8x8-бітового беззнакового множення зі швидкістю 1,375 мкс (при роботі з тактовою частотою $f_{CPU}=8$ МГц). Основні характеристики деяких модифікацій мікроконтролерів ST7 наведено в табл.6.14.

Таблиця 6.14. Основні характеристики деяких модифікацій мікроконтролерів ST7

Характеристика	ST7LITE20	ST7LITE25	ST7LITE29
Пам'ять програм, байт	8К		
ОЗП, байт	384 (128)		
EEPROM даних, байт	-	-	256
Периферійні пристрої	Lite-таймер, сторожовий таймер (Watchdog), таймер з автоперезавантаженням, SPI, 10-розрядний АЦП	Lite-таймер, сторожовий таймер (Watchdog), таймер автоперезавантаження з вхідним тактуванням 32 МГц, SPI, 10-розрядний АЦП	
Напруга живлення	(2,4-5,5) В		
Частота процесора	До 8 МГц (із зовнішнім тактуванням - до 16МГц)	До 8 МГц (із зовнішнім тактуванням - до 16МГц і зовнішнім 1МГц RC генератором з 1% підстроюванням)	
Допустима температура середовища	(-40... +85)°C		
Тип корпусу	SO20 300°, DIP20		

Розглянемо більш докладно мікроконтролер ST7Lite2. Він має:

- супервізор напруги живлення;
- 15 багатофункціональних двонапрямлених ліній введення-виведення, з них 7 виходів з підвищеною навантажною здатністю (20 мА);
- 10 векторів переривань;
- вектори TRAP (покроковий режим) і RESET (скидання);
- 10-бітний ADC (вхідна напруга до 430 мВ та до 5В; 7 входів);
- вбудований модуль відладки *Debug*.

МК може працювати в 5 режимах енергоспоживання – *Halt, Active-Halt, Wait and Slow, Auto Wake Up From Halt*.

Структурна схема мікроконтролера ST7LITE29 наведена на рис. 6.28.

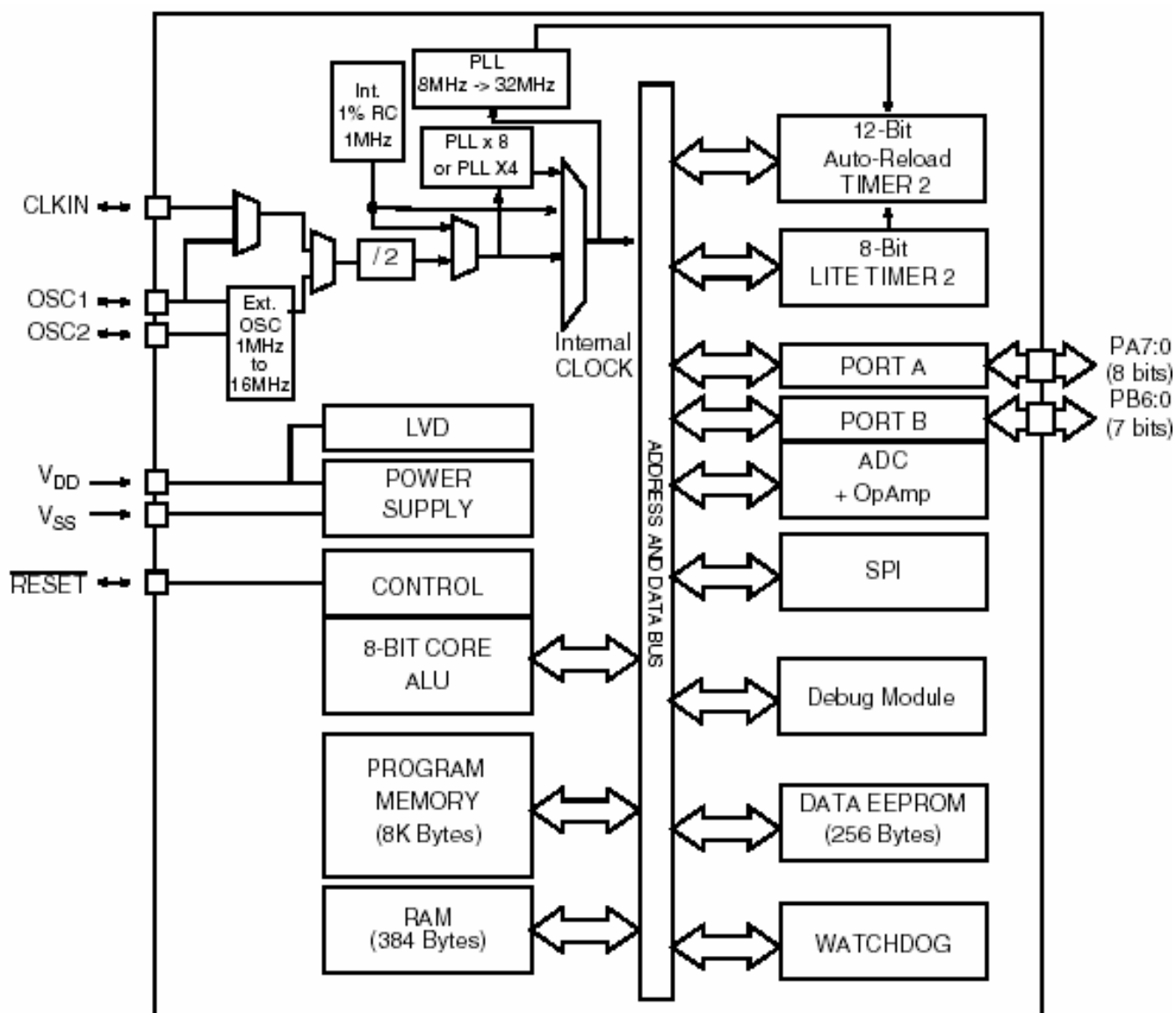


Рисунок 6.23 - Структурна схема мікроконтролера ST7

Контролер містить: OSC – генератор 1-16 МГц, стабілізований кварцевим резонатором; 1% RC-генератор; PLL (Phase Locked Loop) – схему фазового автопідстроювання для множення частоти; LVD – супервізор напруги живлення; Power Supply – джерело напруги; 8 bit Core – ядро МК; Program Memory – пам'ять програм (FLASH – 8 кбайт); RAM – ОЗП даних (384 байт); 12 bit Auto Reload Timer 2 – таймер з автозавантаженням; 8 bit Lite Timer 2 – два Lite таймери; Port A,B – порти введення/виведення; ADC – АЦП; SPI – синхронний послідовний порт; Debug – модуль відладки; DATA EEPROM – енергонезалежне ОЗП (256 байт); WATCDOG – сторожевий таймер.

Умове графічне позначення МК ST7LITE2 наведено на рис.6.24.

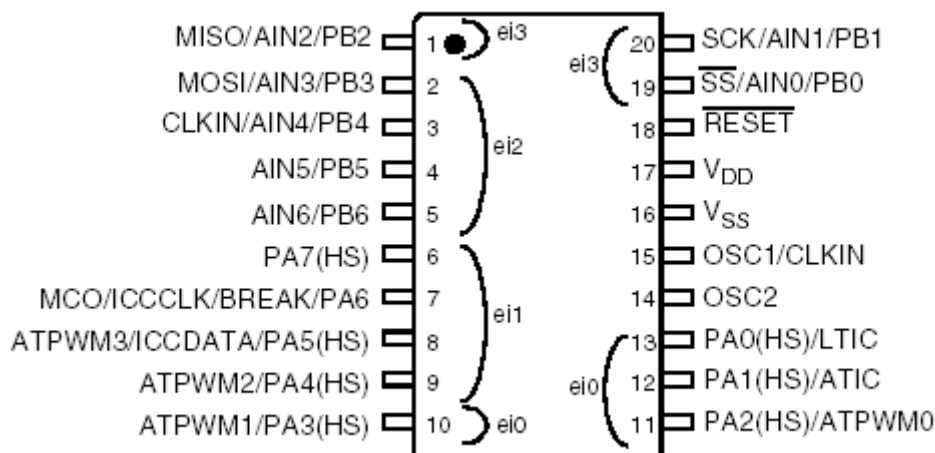


Рисунок 6.24 - Найменування виводів для типу корпусу DIP20

Найменування виводів МК ST7LITE2 наведено у табл. 6.15.

Таблиця 6.15. Найменування виводів мікроконтролера

№ виводу		Найменування виводу	Рівень	Конфігурація портів / керування						Основна функція (після скидан-ня)	Альтернативна функція			
SO20	DIP20			Тип	Вх.	Вих.	Вх.					Вих.		
							float	Wpu	Int			Ana	OD	PP
1	16	VSS	S										земля	
2	17	VDD	S										основне джерело живлення	
3	18	RESET	I/O	CT			X			X			немасковане переривання з вищим пріоритетом	
4	19	PB0/AIN0/SS	I/O	CT	X				X	X	X	Порт B0	аналоговий вхід АЦП 0 / вибір веденого (Slave) пристрою інтерфейсу SPI (активний рівень – 0)	
5	20	PB1/AIN1/SCK	I/O	CT	X				X	X	X	Порт B1	Аналоговий вхід АЦП 1 / тактовання SPI	
6	1	PB2/AIN2/MISO	I/O	CT	X				X	X	X	Порт B2	Аналоговий вхід АЦП 2 / режим SPI: Master In/ Slave Out	
7	2	PB3/AIN3/MOSI	I/O	CT	X				X	X	X	Порт B3	Аналоговий вхід АЦП 3 / режим SPI: Master Out / Slave In	
8	3	PB4/AIN4/CLKIN	I/O	CT	X				X	X	X	Порт B4	Аналоговий вхід АЦП 4 / вхідні імпульси тактування	
9	4	PB5/AIN5	I/O	CT	X				X	X	X	Порт B5	Аналоговий вхід АЦП 5	
10	5	PB6/AIN6	I/O	CT	X				X	X	X	Порт B6	Аналоговий вхід АЦП 6	

11	6	PA7	I/O	CT	HS	X	ei1		X	X	Порт А7	
12	7	PA6/MCO/ ICCCLK/BREAK	I/O	CT	HS	X	ei1		X	X	Порт А6	Вихід основного тактування / тактування внутрішньосхемного програмування / зовнішній сигнал останову
13	8	PA5/ATPWM3/ ICCDATA	I/O	CT	HS	X	ei1		X	X	Порт А5	ШІМ3 таймера автоперезавантаження/дані внутрішньосхемного програмування
14	9	PA4/ATPWM2	I/O	CT	HS	X			X	X	Порт А4	ШІМ2 таймера автоперезавантаження
15	10	PA3/ATPWM1	I/O	CT	HS	X	ei0		X	X	Порт А3	ШІМ1 таймера автоперезавантаження
16	11	PA2/ATPWM0	I/O	CT	HS	X			X	X	Порт А2	ШІМ0 таймера автоперезавантаження
17	12	PA1/ATIC	I/O	CT	HS	X			X	X	Порт А1	Вхід захоплення таймера автоперезавантаження
18	13	PA0/LTIC	I/O	CT	HS	X			X	X	Порт А0	Вхід захоплення Lite-таймера
19	14	OSC2	O									Вихід інвертування генератора резонатора
20	15	OSC1/CLKIN	I									Вхід інвертування генератора резонатора / Вхід зовнішнього тактування

Примітка:

Тип : I – вхідний вивід, O – вихідний вивід, S – вивід живлення;

Рівень: Ct – КМОП 0,3VDD (0,7VDD з вхідною схемою тригера); HS – вихідний рівень: 20mA з покращеним тепловідводом;

Конфігурація портів / керування: float – вхід з плаваючою точкою, wpi – вхід із підтягувальним резистором, int - вхід переривання, ana – вхід аналоговий вхід; OD – вихід з відкритим стоком, PP – вихід двотактний; SPI – послідовний інтерфейс.

Ядро МК. До ядра МК входять: 8-бітний арифметико-логічний пристрій (АЛП) з вбудованим блоком множення 8-розрядних чисел та 6 внутрішніх регістрів:

- Акумулятор (А) – 8-розрядний регістр загального призначення, який використовується для фіксації операндів і результатів арифметичних і логічних обчислень, а також для керування даними.,
- X і Y – 8-розрядні індексні регістри, які у режимах індексної адресації використовуються для створення або ефективної адреси, або ділянок для тимчасового зберігання даних, з якими проводяться маніпуляції,
- Програмний лічильник (PC) – 16-розрядний регістр, що містить адресу наступної команди. Він складається з двох 8-розрядних регістрів: PCL (Program Counter Low – молодші

розряди програмного лічильника) і PCH (Program Counter High – старші розряди програмного лічильника). При скиданні в нього завантажується значення FFFEH.

- Показчик вершини стека (SP) – 16-розрядний регістр, який вказує на наступне вільне місце розташування даних в стеку. Він декрементується після того, як дані розміщені в стеку, і інкрементується перед витягненням даних із стеку. Оскільки стек має глибину 128 байт, 9 старших біт встановлюються примусово апаратно. Досягши нижньої межі, показчик стека повертається до верхньої межі стека без вказівки на переповнення стека. Раніше збережена інформація буде загублена. При скиданні в нього завантажується значення 01FFh.

- Регістр коду стану (CCR) - 8-розрядний регістр, який містить маску переривання та 4 прапорці – результат виконання команд.

Формат регістру коду стану (CCR – *Condition Code Register*) наведено на рис. 6.25.

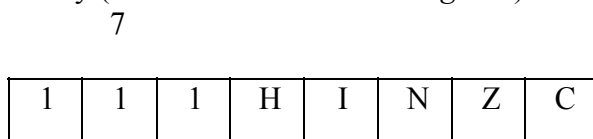


Рисунок 6.25 - Формат регістру коду стану

На рис.6.25 позначено:

H (Half carry) - напівперенесення. Цей біт встановлюється апаратно, якщо під час виконання операції було перенесення із 3-го розряду результату. Значення цього біта може бути перевірене командами JRH або JRNH. Цей біт використовується в підпрограмах, що використовують BCD-арифметику;

I (Interrupt mask) - маска переривання. Біт встановлюється апаратно при вході в режим переривання або програмно для заборони всіх переривань, окрім програмного TRAP-переривання. Скидається програмно. При $I=1$ переривання заборонено. Значення цього біта може бути перевірене командами JRM або JRNM, управляється командами RIM, SIM і IRET;

N (Negative) – негативний результат. Встановлюється, коли старший біт результату дорівнює 1. Він представляє собою знак результату попередньої арифметичної або логічної операції, або операції з даними. Це, по суті, копія 7-го біта результату. Якщо біт дорівнює 0, то результат попередньої операції додатний або дорівнює нулю, якщо 1 - то від'ємний (старший біт результату = 1), Значення цього біта може бути перевірене командами JRM1 або JRPL;

Z (Zero)– нуль. Цей біт встановлюється і скидається апаратно. Встановлюється, якщо результат останньої арифметичної або логічної операції, або операції з даними дорівнює нулю. Значення цього біта може бути перевірене командами JREQ або JRNE;

C (Carry/borrow) - перенесення/позика. Біт встановлюється і скидається апаратно і програмно. Він показує, що в процесі попередньої арифметичної операції відбулося переповнення або позика із старшого розряду. Значення цього біта може бути перевірене командами JRC або JRNC, змінюватися командами SCF і RCF.

Пам'ять ST7Lite2 є лінійною і займає обсяг від 1 Кбайт до 64 Кбайт. У ST7Lite2 використовується 3 типи пам'яті:

- програмна пам'ять FLASH – 8 Кбайт;
- пам'ять даних (RAM) і стек – 384 байт;
- програмована пам'ять з електричним стиранням (EEPROM) – 256 байт.

Карта пам'яті наведена на рис. 6.26.

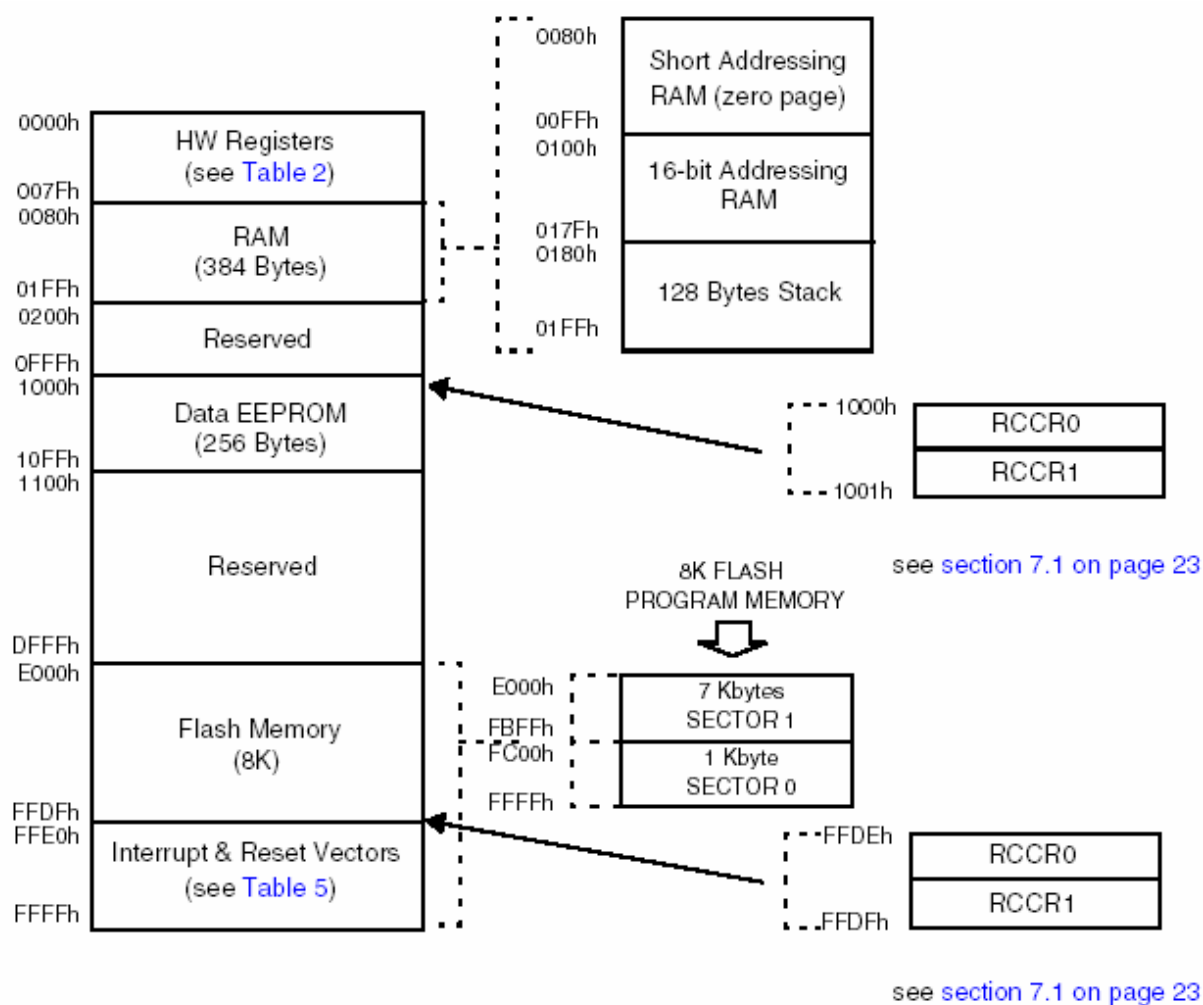


Рисунок 6.25 - Карта пам'яті

Перелік регістрів спеціальних функцій *HW Registers* наведено в табл.6.16.

Таблиця 6.16 Регістри спеціальних функцій

Адреса	Блок	Позначення регістра	Найменування регістра	Статус скидання	Зауваження
0000h 0001h 0002h	Порт А	PADR PADDDR PAOR	Регістр даних порту А Регістр напрямку передачі порту А Регістр опцій порту А	FFh1) 00h 40h	R/W R/W R/W
0003h 0004h 0005h	Порт В	PBDR PBDDR PBOR	Регістр даних порту В Регістр напрямку передачі порту В Регістр опцій порту В	FFh 1) 00h 00h	R/W R/W R/W
0006h 0007h	Зарезервована ділянка (2 байти)				
0008h 0009h 000Ah 000Bh 000Ch	LITE таймер 2	LTCSR2 LTARR LTCNTR LTCSR1 LTICR	Регістр керування/статусу 2LITE-таймера Регістр автоперезавантаження LITE-таймера Рахунковий регістр LITE-таймера Регістр керування/статусу 1 LITE-таймера Регістр вхідного захоплення LITE-таймера	0Fh 00h 00h 0X00 0000h xxh	R/W R/W RO R/W RO

Адреса	Блок	Позначення регістра	Найменування регістра	Статус скидання	Зауваження
000Dh	Таймер автоперезавантаження 2	ATCSR	Регістр керування/статусу таймера	0X00 0000h	R/W
000Eh		CNTRH	Рахунковий регістр (старші розряди)	00h	RO
000Fh		CNTRL	Рахунковий регістр (молодші розряди)	00h	RO
0010h		ATRH	Регістр автоперезавантаження (ст.розр)	00h	R/W
0011h		ATRL	Регістр автоперезавантаження (мол.розр.)	00h	R/W
0012h		PWMCR	Регістр керування виходу ШІМ	00h	R/W
0013h		PWM0CSR	Регістр керування/статусу ШІМ 0	00h	R/W
0014h		PWM1CSR	Регістр керування/статусу ШІМ 1	00h	R/W
0015h		PWM2CSR	Регістр керування/статусу ШІМ 2	00h	R/W
0016h		PWM3CSR	Регістр керування/статусу ШІМ 3	00h	R/W
0017h		DCR0H	Регістр робочого циклу ШІМ 0 (ст.)	00h	R/W
0018h		DCR0L	Регістр робочого циклу ШІМ 0 (мл.)	00h	R/W
0019h		DCR1H	Регістр робочого циклу ШІМ 1 (ст.)	00h	R/W
001Ah		DCR1L	Регістр робочого циклу ШІМ 1 (мл.)	00h	R/W
001Bh		DCR2H	Регістр робочого циклу ШІМ 2 (ст.)	00h	R/W
001Ch		DCR2L	Регістр робочого циклу ШІМ 2 (мл.)	00h	R/W
001Dh		DCR3H	Регістр робочого циклу ШІМ 3 (ст.)	00h	R/W
001Eh		DCR3L	Регістр робочого циклу ШІМ 3 (мл.)	00h	R/W
001Fh		ATICRH	Регістр захоплення входу ст.)	00h	RO
0020h		ATICRL	Регістр захоплення входу (мл)	00h	RO
0021h		TRANCR	Регістр керування передачею	01h	R/W
0022h	BREAKCR	Регістр керування перерви в роботі	00h	R/W	
0023h - 002Dh	Зарезервована ділянка (11 байт)				
002Eh	WDT таймер	WDGCR	Регістр керування сторожовим таймером (Watchdog)	7Fh	R/W
0002Fh	Флеш-пам'ять	FCSR	Регістр керування/статусу флеш-пам'яті	00h	R/W
00030h	EEPROM	EECSR	Регістр керування/статусу EEPROM	00h	R/W
0031h	SPI	SPIDR	Регістр введення/виведення. даних SPI	xxh	R/W
0032h		SPICR	Регістр керування SPI	0xh	R/W
0033h		SPICSR	Регістр керування /статусу SPI	00h	R/W
0034h	АЦП	ADCCSR	Регістр керування/ статусу АЦП	00h xxh	R/W
0035h		ADCDRH	Регістр даних АЦП (ст.)	0xh	RO
0036h		ADCDDL	Регістр даних АЦП(мл.) /контроль підсилювача АЦП		R/W
0037h	ІТС (контролер зовнішнього переривання)	EICR	Регістр керування зовнішніми перериваннями	00h	R/W

Адреса	Блок	Позначення регістра	Найменування регістра	Статус скидання	Зауваження
0038h	МСС (контроллер основного процессора)	MCCSR	Регістр керування/статусу синхроімпульсів	00h	R/W
0039h 003Ah	Тактування і скидання	RCCR SICSR	Регістр контроллера RC генератора Регістр керування/статусу цілісності системи	FFh 0000 0XX0h	R/W R/W
003Bh	Зарезервована ділянка (1 байт)				
003Ch	ІТС (вибір зовнішнього переривання)	EISR	Регістр вибору зовнішнього переривання	0Ch	R/W
003Dh to 0048h	Зарезервована ділянка (12 байт)				
0049h 004Ah	AWU	AWUPR AWUCSR	Регістр передлічильника AWU Регістр керування/статусу AWU	FFh 00h	R/W R/W
004Bh 004Ch 004Dh 004Eh 004Fh 0050h	Модуль відлагодження	DMCR DMSR DMBK1H DMBK1L DMBK2H DMBK2L	Регістр керування Регістр статусу Регістр 1 точок останову (ст.) Регістр 1 точок останову (мл.) Регістр 2 точок останову (ст.) Регістр 2 точок останову (мл.)	00h 00h 00h 00h 00h 00h	R/W R/W R/W R/W R/W R/W
0051h - 007Fh	Зарезервована ділянка (47 байт)				

Примітки: х=не визначено, R/W- читання/запис, RO-тільки для читання

У пам'яті програм зберігаються основна прикладна програма і підпрограми, які визначають всі необхідні завдання 8-розрядного ядра. Пам'ять програм являє собою флеш-пам'ять із захистом читання, має 10К циклів перепрограмування; містить 2 сектора 0 (1 Кбайт) та 1 (7 Кбайт). В секторі 0 знаходиться карта векторів переривання. Є три режими програмування пам'яті програм:

1. Встановлення пристрою програмування. У цьому режимі сектори 0 і 1 флеш-пам'яті і байти опцій можуть бути запрограмовані або стерті.
2. Внутрішньосхемне програмування (*ICP - In-Circuit Programming*). У цьому режимі сектори 0 і 1 флеш-пам'яті і байти опцій можуть бути запрограмовані або стерті без видалення МК із плати.
3. Програмування усередині додатку (*IAP - In-Application Programming*). У цьому режимі сектор 1 флеш-пам'яті може бути запрограмований або стертий без видалення МК із плати і без припинення роботи з додатком.

Розмір нульового сектора настроюється по спеціальному байту. Існує 2 види захисту пам'яті: захист від прочитування (для запобігання порушення авторських прав) і захист від запису/стирання, вони можуть бути застосовані незалежно.

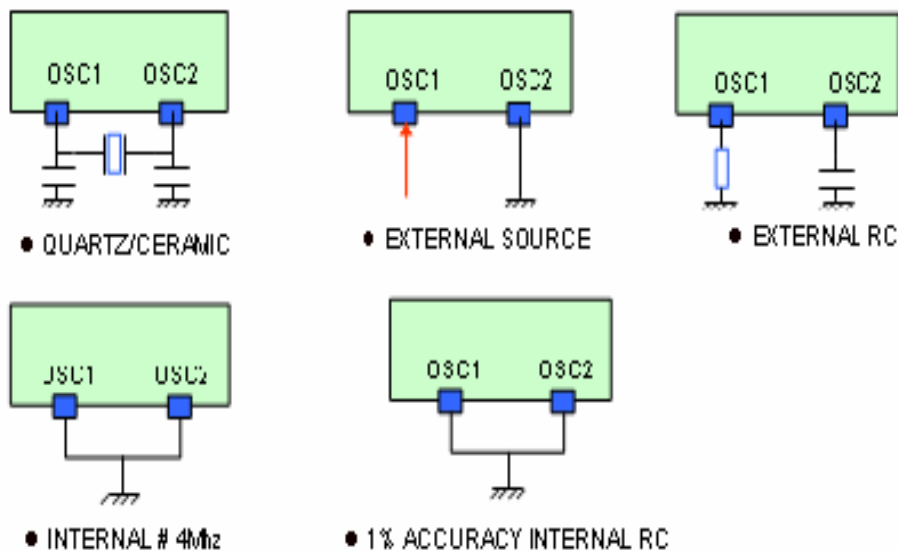
Пам'ять даних. Внутрішня оперативна пам'ять RAM (384 байт) призначена для зберігання тимчасових результатів обчислень або контексту стану МК при перериванні. ОЗП містить 2 сторінки: 0 та 1. При цьому сторінка 0 ОЗП (ОЗП 0) має короткий спосіб адресації, більш швидкий і з меншою кількістю розрядів (8-розрядна адреса). Сторінка 1 ОЗП (ОЗП 1) використовує спосіб адресації словом (16-розрядна адреса). Апаратні регістри також адресуються з використанням короткого способу адресації.

Стек розташований в ОЗП з початковою адресою 1FF. Оскільки стек заповнюється спочатку від вищих адрес, то у випадку, якщо додаток не вимагає повного вмісту стека, верхня частина стека може використовуватися як ОЗП.

Енергонезалежна пам'ять даних з електричним стиранням (256 байт) використовується для збереження констант та таблиць. В ній розташовані регістри керування параметрами RC – генератора RCCR0 та RCCR1. Основні особливості EEPROM наступні:

1. Програмування за один цикл до 32 байт;
2. Пакетні цикли стирання і програмування;
3. Внутрішній контроль тривалості загального циклу програмування;
4. Керування режимом очікування (Wait-mode);
5. Захист від прочитування.

Тактовий генератор В мікроконтролері використовуються наступні джерела



тактування (рис. 6.31):

Рисунок 6.31 - Джерела тактування

- внутрішній генератор, стабілізований кристалічним/керамічним резонатором, (рис. 6.31,а);
- зовнішні тактові сигнали (рис. 6.31,б);
- внутрішній RC-генератор (рис. 6.31,в);
- внутрішній генератор 4 МГц (рис. 6.31,г);
- внутрішній 1% RC генератор (рис. 6.31,д).

Тактовий генератор формує також 32 МГц імпульси для таймера Auto-reload та має в своєму складі опціональну схему фазової автопідстройки частоти (PLL) для отримання 4 або 8 МГц шляхом множення частоти на 8 або 4.

Керування тактовим генератором здійснюється за допомогою регістру керування/статусу синхроімпульсів MCCSR (MAIN CLOCK CONTROL/STATUS REGISTER) та RCCR (RC CONTROL REGISTER) - регістр керування RC-генератора.

Структура регістру MCCSR наведена на рис. 6.32, де позначено:

- MCO (Main Clock Out - головний вихід тактової частоти). При MCO=1 головний вихід тактової частоти під'єднаний до виводу MCO;

- SMS (Slow mode select - повільний режим) Біт SMS задає частоту вхідних синхроімпульси f_{OSC} (при SMS=0) або $f_{OSC}/32$ (при SMS=1).

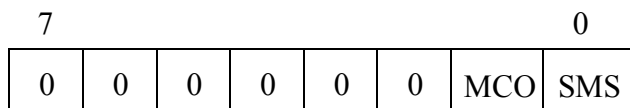


Рисунок 6.32 - Формат регістра MCCR

Формат регістра RCCR подано на рис.6.33, де CR[7:0] - біти визначення частоти RC-генератора.

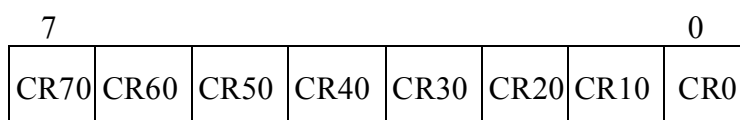


Рисунок 6.33 - Формат регістра RCCR

Якщо в регістр RCCR занести 00H, то RC-генератор матиме максимальну частоту, якщо – FFh, то мінімальну частоту.

Скидання RESET можна здійснити трьома шляхами:

1. Зовнішнім імпульсом скидання RESET;
2. Внутрішнім скиданням при падінні напруги живлення нижче рівня $V_{IT-(LVD)}$, рис.6.34;
3. Внутрішнім скиданням по завершенні часу очікування сторожового таймера.

Вивід скидання є двоспрямованим і має вбудований малопотужний резистор навантаження. Для скидання необхідно подати низький рівень на вивід скидання RESET. Час скидання складає 30 мкс. Розглянемо більш докладно внутрішнє скидання за сигналами LVD та AVD (див. рис. 6.34).

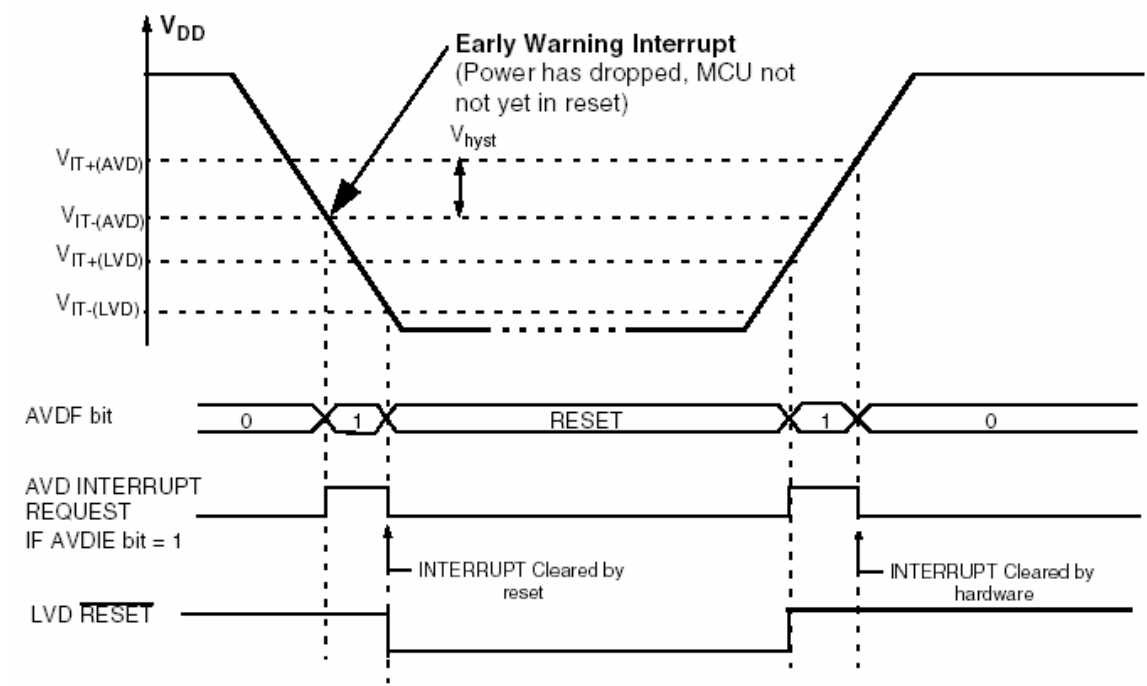


Рисунок 6.34 – Генерація сигналів RESET при падінні напруги живлення

Супервізор напруги LVD (LVD-датчика низької напруги) реалізує скидання, якщо напруга живлення V_{DD} опускається нижче встановленого значення. Порогова напруга формується відповідним опційним бітом, і її рівень може бути низьким, середнім або високим.

Функція допоміжного детектора напруги (AVD) заснована на аналоговому порівнянні напруги, що подається на мікроконтролер, із заданим значенням напруги живлення V_{DD} (U_{AVD}). Вихід порівняння AVD може бути прочитаний у реальному часі за

допомогою відповідного програмного додатку, завдяки наявності біта AVDF в регістрі SICSR (рис. 6.35). Цей біт доступний тільки для читання.

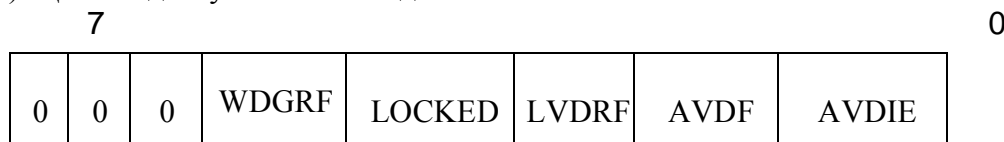


Рисунок 6.35. Формат регістра SICSR

Функція AVD може використовуватися тільки, якщо встановлено дозвіл відповідним бітом LVD. Допоміжний детектор напруги (AVD) може викликати переривання. У разі зниження напруги живлення мікроконтролера нижче значення $V_{IT-(AVD)}$, виникає переривання раннього сповіщення AVD, внаслідок чого програма завершується безпечно, без скидання мікроконтролера.

На рис. 6.35 позначено: WDGRF – прапор скидання сторожового таймера. Цей біт вказує, що останнє скидання було вироблене сторожовим таймером. Він встановлюється апаратно (за сигналом сторожового таймера), а скидається програмно або апаратно – LVD скиданням (щоб гарантувати нульове значення прапорця WDGRF при запуску мікроконтролера). Спільно з бітом LVDRF біт WDGRF визначає джерело скидання (табл.6.17).

Таблиця 6.17. - Джерела скидання

Джерело скидання	LVDRF	WDGRF
Зовнішнє скидання на вивід RESET	0	0
По сторожовому таймеру	0	1
Скидання LVD	1	X

Біт LOCKED - прапор сповіщення про блокування фазового автопідстроювання. Цей біт скидається і встановлюється апаратним забезпеченням. Установка відбувається автоматично, коли частота фазового автопідстроювання досягне робочої частоти. Якщо біт дорівнює 0, то фазове автопідстроювання не блоковане, якщо -1, то фазове автопідстроювання блоковане.

Біт LVDRF – прапор скидання LVD. Цей біт вказує, що останнє скидання було вироблене блоком LVD. Біт встановлюється апаратно (LVD скидання) і очищується програмно. Коли LVD відключений відповідним бітом дозволу, значення біта LVDRF невизначено.

Біт AVDF – прапор детектора напруги. Цей біт використовується тільки для читання, він встановлюється і очищується апаратно. Якщо біт AVDIE встановлений і запит на переривання вироблений, тоді біт AVDF встановлений. Якщо AVDF=0, то Vdd дорівнює рівню AVD, якщо AVDF=1, то Vdd нижче за рівень AVD.

Біт AVDIE – прапор дозволу переривання детектором напруги. Цей біт встановлюється і очищується програмно. Це дозволяє перериванню здійснитися, коли прапор AVDF встановлений. Якщо переривання не було здійснено, то прапор автоматично очищується. Якщо AVDIE =0, то AVD переривання заборонені, якщо AVDIE=1, то AVD переривання дозволені.

Переривання

МК ST7 має два різні типи переривань: масковані і немасковані. Немасковане програмне переривання активується командою TRAP і виконується незалежно від стану біта I. Типи переривань і початкові адреси підпрограм їх обробки наведено в табл. 6.18.

Таблиця 6.18. Джерела переривань

№	Джерело	Опис	Пріоритет	Адреси підпрограми (вектор)
1	RESET	Скидання	Найвищий пріоритет	FFFEH – FFFFH
2	TRAP	Програмне переривання		FFFC – FFFDH
3	AWU	Автопробудження від переривання		FFFA – FFFBH
4	ei0	Зовнішнє переривання 0		FFF8H – FFF9H
5	ei1	Зовнішнє переривання 1		FFF6H – FFF7H
6	ei2	Зовнішнє переривання 2		FFF4H – FFF5H
7	ei3	Зовнішнє переривання 3		FFF2H – FFF3H
8	LITE TIMER	LITE таймер RTC2 – переривання за переповненням		FFF0H – FFF1H
9		Не використовується		FFEEH – FFEFH
10	SI	AVD переривання		FFECH – FFEDH
11	AT TIMER	AT таймер, який порівнює вихідне переривання з вхідним		FFEAH – FFE9H
12		AT таймер – переривання по переповненню		FFE8H – FFE7H
13	LITE TIMER	LITE таймер – захват вхідного переривання		FFE6H – FFE5H
14		LITE таймер RTC1 – переривання по переповненню		FFE4H – FFE3H
15	SPI	SPI переривання введення/виведення		FFE2H – FFE1H
16		Не використовується	FFE0H-FFE1H	

Перед виконанням переривання в стек записується: адреса команди, на яку треба повернути після обробки переривання (зміст регістра PC), регістри X, A, CC. Зміну всіх параметрів, що зв'язані з перериваннями, рекомендують робити при заборонених перериваннях. Програма обробки переривання повинна закінчитися командою IRET, яка відновлює значення збережених регістрів із стеку.

Як видно з табл. 6.18, МК може обробляти 4 зовнішні переривання, причому можна програмно задати як чутливість (тип сигналу) переривання, так і номер виводу BIC МК. Тип сигналу задається за допомогою регістра керування зовнішніми перериваннями EICR (EXTERNAL INTERRUPT CONTROL REGISTER). Формат регістра EICR подано на рис.6.36.

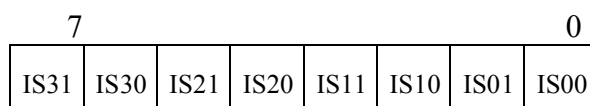


Рисунок 6.36 - Формат регістра EICR

Біти IS0[1:0], IS1[1:0], IS2[1:0], IS3[1:0] визначають чутливість зовнішніх переривань 0-3 відповідно (табл. 6.19).

Таблиця 6.19. Чутливість зовнішнього переривання

Isx1	Isx0	Чутливість зовнішнього переривання
0	0	По спаду і низькому рівню

0	1	Тільки по фронту
1	0	Тільки по спаду
1	1	По спаду і по фронту

Номер виводу BIC МК задається за допомогою регістру вибору зовнішнього переривання EISR (EXTERNAL INTERRUPT SELECTION REGISTER), формат якого подано на рис.6.37.

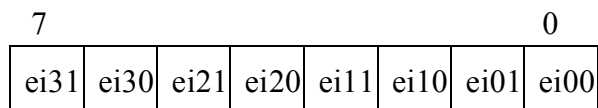


Рисунок 6.37. Формат регістра EISR

Біти ei3[1:0] задають лінію введення/виведення для зовнішнього переривання 3 відповідно до табл. 6.20.

Таблиця 6.20. Вибір ліній введення/виведення для зовнішнього переривання 3

ei31	ei30	Лінії введення/виведення
0	0	PB0*
0	1	PB1
1	0	PB2

* Стан скидання

Біти ei2[1:0] задають лінію введення/виведення для зовнішнього переривання 2 відповідно до табл. 6.21.

Таблиця 6.21. Вибір ліній введення/виведення для зовнішнього переривання 2

ei21	ei20	Введення/виведення
0	0	PB3*
0	1	PB4
1	0	PB5
1	1	PB6

* Стан скидання

Біти ei1[1:0] задають лінію введення/виведення для зовнішнього переривання 1 відповідно до табл. 6.22.

Т

Таблиця 6.22 - Вибір ліній введення/виведення для зовнішнього переривання 1

ei11	ei10	Введення/виведення
0	0	PA4
0	1	PA5
1	0	PA6
1	1	PA7*

* Стан скидання

Біти ei0[1:0] задають лінію введення/виведення для зовнішнього переривання 0 відповідно до табл. 6.23.

Таблиця 6.23. Вибір ліній введення/виведення для зовнішнього переривання 0

ei01	ei00	Введення/виведення
0	0	PA0*
0	1	PA1
1	0	PA2
1	1	PA3

* Стан скидання

Приклад:

```
it_BP1:          ; приклад підпрограми, для опрацювання переривання
               clr   Go_on ; скидання біту Go_on          IRET
```

```
; -----INTERRUPT SUB-ROUTINES SECTION -----
.dummy        iret          ; Empty subroutine
.sw_rt        iret
.ext0_rt      call it_BP1:
               iret
.ext1_rt      iret
.ext2_rt      iret
.ext3_rt      iret
.avd_rt       iret
.artim_oc_rt  iret
.artim_ovf_rt iret
.ltim_ic_rt   iret
.ltim_rtc_rt  iret
.spi_rt       iret
```

```
;МАПА ВЕКТОРІВ ПЕРЕРИВАНЬ -----
;сегмент 'vectit' залежить від мапування пам'яті пристрою
```

;Нижче наведено мапу переривань для сімейства ST7Lite2.

```
segment 'vectit'
               DC.W    dummy _rt ; FFE0-FFE1h location
.spi_it       DC.W    spi_rt      ; FFE2-FFE3h location
.ltim_rtc     DC.W    ltim_rtc_rt  ; FFE4-FFE5h location
.ltim_ic      DC.W    ltim_ic_rt   ; FFE6-FFE7h location
.artim_ovf    DC.W    artim_ovf_rt ; FFE8-FFE9h location
.artim_oc     DC.W    artim_oc_rt  ; FFEA-FFEBh location
.avd_it       DC.W    avd_rt       ; FFEC-FFEDh location
               DC.W    dummy_rt    ; FFEE-FFEFh location
               DC.W    dummy _rt   ; FFF0-FFF1h location
.ext3_it      DC.W    ext3_rt      ; FFF2-FFF3h location
.ext2_it      DC.W    ext2_rt      ; FFF4-FFF5h location
.ext1_it      DC.W    ext1_rt      ; FFF6-FFF7h location
.ext0_it      DC.W    ext0_rt      ; FFF8-FFF9h location
               DC.W    dummy _rt    ; FFFA-FFFBh location
.softit      DC.W    sw_rt        ; FFFC-FFFDh location
.reset        DC.W    main         ; FFFE-FFFFh location
```

; Остання строка посилається на першу строку. Це потрібно для того, щоб компілятор знав межі зони коду.

Режими енергоспоживання

В МК ST7, крім робочого режиму (RUN), існують 5 режимів енергоспоживання:

1. Повільний (Slow);

2. Чекання (Wait);
3. Активна Зупинка (Active Halt);
4. Автоматичне пробудження з режиму зупинки (AWUFH - Auto Wake Up From Halt);
5. Зупинка (Halt).

Після скидання МК переходить у робочий режим RUN та запускає CPU. Після цього програмним шляхом можна встановити один з режимів зменшеного енергоспоживання.

Повільний режим Slow служить для зменшення споживання енергії шляхом зменшення частоти внутрішніх синхроімпульсів МК або для пристосування внутрішньої тактової частоти до зменшеної напруги живлення. Перехід в цей режим здійснюється встановленням біту SMS в регістрі MCCSR (див. рис.6.32) в одиницю. Вихід з режиму здійснюється скиданням цього біту. У повільному режимі частота генератора ділиться на 32, тому CPU і периферійні пристрої працюють на низькій частоті.

Режим чекання Wait знижує енергоспоживання мікроконтролера шляхом зупинки CPU. Цей режим викликається командою WFI. Всі периферійні пристрої залишаються активними. Під час режиму чекання біт I регістра CCR (див. рис.6.25) очищується, тобто всі переривання дозволено. Вміст регістрів і пам'яті залишаються незмінними. Мікроконтролер залишається в цьому режимі, поки не буде вироблене переривання або скидання, після чого у лічильнику команд встановлюється початкова адреса переривання або скидання. Різновидом цього режиму є режим SLOW WAIT. Для переходу в нього біт SMS регістру MCCSR встановлюють в 1, тим самим зменшуючи частоту роботи в 32 рази. Вихід з цього режиму відбувається також за перериванням або скиданням.

Режим зупинки Halt – режим з найнижчим енергоспоживанням мікроконтролера. Цей режим викликається командою HALT з попередньо скинутим бітом AWUEN в регістрі AWUCSR (див. рис.6.38). У режимі зупинки основний генератор вимкнений. Всі периферійні пристрої зупиняються, окрім тих, які одержують синхроімпульси від іншого тактового, зовнішнього або допоміжного генератора. Робота сторожового таймера в режимі зупинки дозволяється або забороняється бітом WDGHALT байта опцій 1 у енергонезалежній пам'яті.

Вихід з режиму може здійснюватися або за зовнішнім перериванням, або скиданням МК. При цьому генератор виконує 256 або 4096 тактів затримки для стабілізації роботи генератору, після чого CPU відновлює режим обслуговування вектора переривання або обирає вектор скидання, який вивів його із стану зупинки. При переході до режиму зупинки біт I регістра CCR (див. рис.6.25) примусово встановлюється в 0, що дозволяє переривання. Тому, якщо переривання незавершене (перехід в режим відбувся під час виконання програми обслуговування переривання), то мікроконтролер вийде із стану зупинки негайно.

Режим активної зупинки Active Halt має низький режим споживання енергії, при цьому мікроконтролер працює із синхроімпульсами реального часу. Введення в цей режим аналогічний введенню в режим HALT. При цьому для того щоб вибрати, в який з режимів входить - активної зупинки або зупинки – необхідно перед переходом до режиму встановити біти регістрів LTCSR та ATCSR згідно з табл. 6.24. Мікроконтролер може вийти з режиму активної зупинки за перериванням або скиданням. При виході з режиму Active Halt за скиданням спочатку проходить 256 або 4096 циклів затримки запуску CPU, після чого CPU відновлює режим шляхом завдання вектору скидання. При виході за перериванням CPU негайно відновлює режим для виконання підпрограми переривання, яке вивело його з режиму активної зупинки.

Таблиця.6.24 - Вибір режиму активної зупинки

LTCSR1 TWIE BIT	ATCSR OVFIE BIT	ATCSR SK1 BIT	ATCSR SK0 BIT	РЕЖИМ
--------------------	--------------------	------------------	------------------	-------

0	X	X	0	режим активної зупинки відключений
0	0	X	X	
1	X	X	X	режим активної зупинки включений
X	1	0	1	

Режим „Автоматичне пробудження з режиму зупинки (AWUFH)”. Цей режим викликається командою HALT з попередньо встановленим бітом AWUEN в регістрі AWUCSR (рис. 6.38), заборонаю режиму *Active Halt* (біти регістрів LTCSR та ATCSR скинуто згідно з табл.6.24) та заданням часу знаходження в режимі зупинки. В режимі AWUFH головний генератор МК вимкнений, але працює AWU RC-генератор з частотою f_{AWU_RC} . Час знаходження в режимі зупинки задається шляхом запису коефіцієнта ділення імпульсів f_{AWU_RC} в регістрі AWUPR (рис.6.39). По закінченню заданого часу встановлюється прапор AWUF в регістрі AWUCSR (рис. 6.38), і генерується переривання, яке виводить мікроконтролер з режиму зупинки.

Зауважимо, що мікроконтролер може вийти з режиму AWUFH за допомогою будь-якої переривання, яке і виводить з режиму зупинки або скиданням. В цьому режимі допускається робота сторожового таймера, який також може генерувати переривання для виходу з режиму AWUFH. Сумісність роботи сторожового таймера з режимом AWUFH задається відповідним байтом WDGHALT байта опцій 1 у енергонезалежній пам'яті.

Формат регістра AWUCSR подано на рис.6.38.

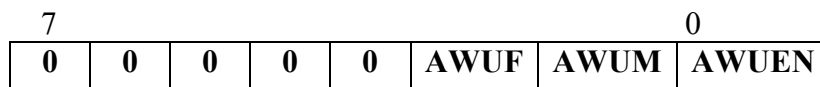


Рисунок 6.38. Формат регістра AWUCSR

На рис. 6.38 позначено:

Біт AWUF (Auto Wake Up Flag) – прапор режиму AWUFH. Цей біт виставляється в 1 апаратно, коли модуль AWU генерує переривання і скидається програмно при читанні AWUCSR.

Біт AWUM (Auto Wake Up Measurement) – біт вимірювання. Цей біт запускає AWU RC генератор і під'єднує його вихід до входу лічильника 12-бітного автоматично перезавантажувального таймера. Це дозволяє використовувати таймер як вимірник відхилення частоти AWU RC генератора і потім вирівняти це відхилення шляхом введення правильних значень в AWUPRE регістр. Якщо біт = 0, то вимірювання відключене, якщо біт =1, то вимірювання включене.

Біт AWUEN (Auto Wake Up From Halt Enabled) – біт дозволу режиму AWUFH. Він встановлюється і скидається програмно. Якщо AWUEN=0, то AWUFH режим вимкнений, якщо AWUEN=1, то AWUFH режим включений.

Формат регістра AWUPR подано на рис.6.39. Значення коефіцієнтів ділення наведено в табл.6.25.

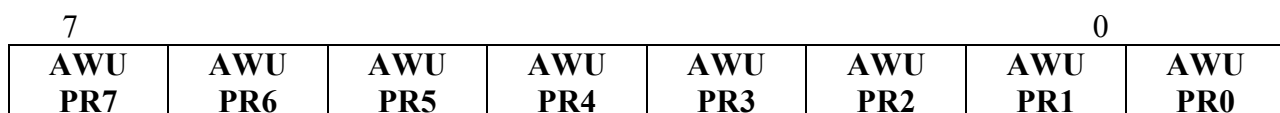


Рисунок 6.39. Формат регістра AWUPR

Таблиця 6.25. Встановлення коефіцієнтів ділення в залежності від бітів регістра AWUPR

AWUPR [7:0]	Значення коефіцієнтів ділення
00h	Заборонено
01h	1
...	...
Feh	254
FFh	255

Вихід із режиму зупинки пояснюється рис.6.40.

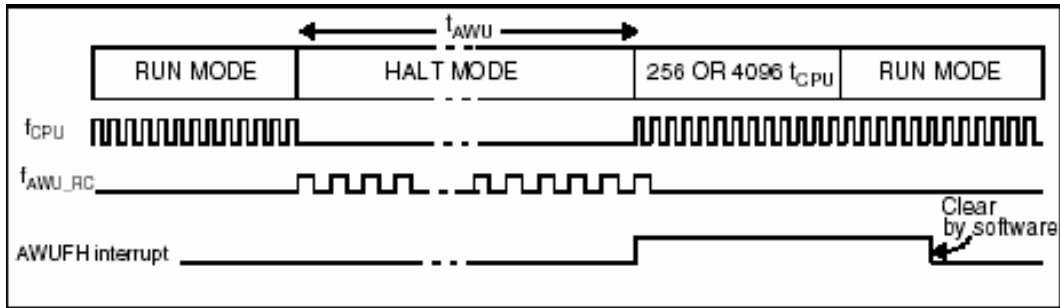


Рисунок 6.40. Діаграми виходу з режиму зупинки

Час роботи режиму зупинки визначається за формулою:

$$t_{AWU} = 64 \cdot AWUPR \cdot \frac{1}{f_{AWURC}} + t_{RCSTRT}$$

Рівні енергоспоживання (від найвищого High до найнижчого Low) розглянутих режимів та можливі переходи з одного до іншого режиму ілюструються рис.6.41.

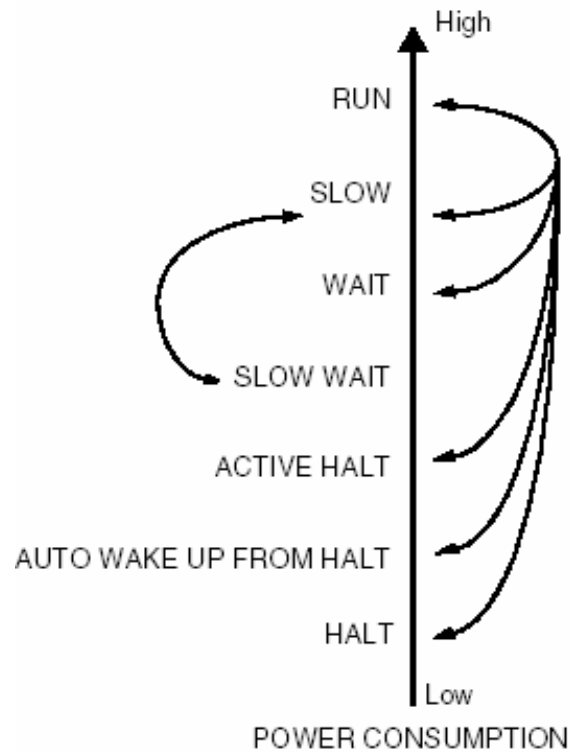


Рисунок 6.41 – Режими енергоспоживання

Порти введення-виведення

Мікроконтролер ST7 має 15 ліній введення/виведення – 7 ліній порту А та 8 ліній порту В. Кожний вивід порту може бути запрограмований на введення або на виведення інформації. Крім того, деякі виводи мають декілька інших функцій, наприклад: зовнішнє переривання, дублювання сигналу введення/виведення для периферійного пристрою на кристалі або для аналогового введення.

Для керування лініями кожний з портів А та В має три регістри (табл. 6.26): регістр даних порту (PADR та PBDR), регістр напряму передачі даних (PADDR, PBDDR), регістр опцій (PAOR, PBOR).

Таблиця 6.26.- Призначення регістрів портів

Порт	Позначення регістрів		Призначення
Порт А	PADR	Port A Data Register	Регістр даних порту А
	PADDR	Port A Data Direction Register	Регістр напряму передачі даних порту А
	PAOR	Port A Option Register	Регістр опцій порту А
Порт В	PBDR	Port B Data Register	Регістр даних порту В
	PBDDR	Port B Data Direction Register	Регістр напряму передачі даних порту В
	PBOR	Port B Option Register	Регістр опцій порту В

Значення біта з номером x (DDR_x) у регістрі даних порту в 0 визначає режим введення, а установка в 1 – режим виведення. Біти регістру опцій DOR_x визначають конфігурацію в режимах введення та виведення згідно табл 6.27.

Таблиця 6.27- Конфігурація ліній портів

Режим конфігурації		DDR	DOR
Введення	Високоімпедансний вхід * (без підтягувального резистора)	0	0
	З підтягувальним резистором	0	1
Виведення	За двотактною схемою (Push-Pull)	1	1
	За схемою з відкритим стоком (Open Drain)	1	0

* Початковий стан

При використанні лінії порту для зовнішнього переривання або для аналогового входу АЦП її треба запрограмувати на режим введення.

Ініціалізація портів полягає у запису даних в регістри напрямку та опцій:

```

init_portA:  ld      A,#INITPADDR      ; 0 - вхід, 1 - вихід відповідного біту регістра
                                                    DDR
              ld      PADDR,A
              ld      A,#INITPAOR     ; 0 – для вхідного виводу - високоімпедансний
                                                    вхід; для вихідного виводу – вихід за схемою з
                                                    відкритим колектором;
                                                    1 – для вхідного виводу – вхід з підтягувальним
                                                    резистором; для вихідного виводу – вихід за
                                                    двотактною схемою
              ld      PAOR,A
ret                                                    ; повернення до головної програми
    
```

```

init_portB:  ld    A,#INITPBDDR    ; 0 - вхід, 1 - вихід відповідного біту регістра
              DDR
              ld    PBDDR,A
              ld    A,#INITPBOR    ; 0 – для вхідного виводу - високоімпедансний
              ; для вихідного виводу – вихід за схемою з
              відкритим колектором;
              ; 1 – для вхідного виводу – вхід з підтягувальним
              резистором; для вихідного виводу – вихід за
              двотактною схемою
              ld    PBOR,A
              ret                ; повернення до головної програми

```

Після цього можна виконувати введення та виведення через порти даних. Підпрограма запису в порт A має вигляд:

```

write_portB: ld    A,portB_TX    ; Завантаження значення вмісту комірки пам'яті
              portB_TX в акумулятор
              ld    PBDDR,A      ; завантаження вмісту регістра DR порту B з
              акумулятора
              ret                ; повернення до головної програми

```

Підпрограма читання порту A має вигляд:

```

read_portB:  ld    A,PBDR        ; завантаження вмісту регістра PBDR порту B в
              акумулятор
              ld    portB_RX,A   ; Завантаження вмісту акумулятора у комірку
              пам'яті portB_RX
              ret                ; повернення до головної програми

```

Ініціалізація, запис та зчитування порту B здійснюється аналогічно.

Послідовний периферійний інтерфейс SPI

Послідовний периферійний інтерфейс SPI (Serial Peripheral Interface) призначений для обміну даними у послідовному форматі між мікроконтролером і периферійними пристроями або між декількома мікроконтролерами ST7.

Дані для передачі та прийняті дані записуються у регістр введення/виведення даних SPIDR. При обміні даними за інтерфейсом SPI мікроконтролер може працювати як у режимі Master, так і у режимі Slave.

Схема підключення двох ВІС МК по інтерфейсу SPI наведена на рис.6.42.

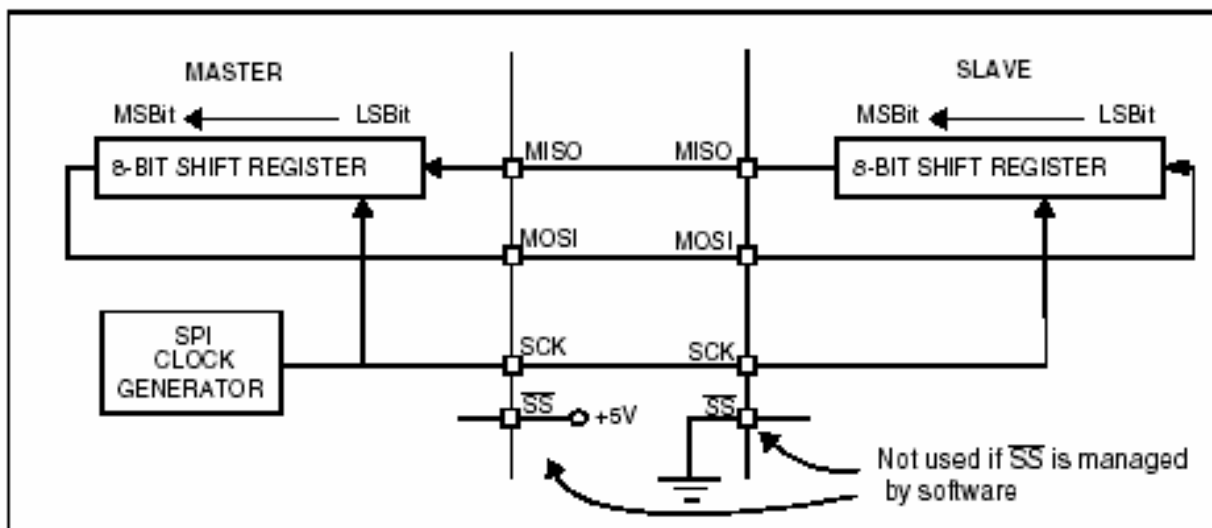


Рисунок 6.42 - Схема підключення двох ВІС МК по інтерфейсу SPI

Обмін за інтерфейсом SPI здійснюється за допомогою 4 виводів ВІС:

- MISO (Master In / Slave Out data) – вхід даних для Master пристрою та вихід для Slave пристрою;
- MOSI (Master Out / Slave In data) – вихід для Master та вхід для Slave пристрою;
- SCK (Serial Clock out by SPI masters and input by SPI slaves) – тактові імпульси (генеруються Master пристроєм і є вхідними для Slave пристрою);
- SS (Slave select) – вибір пристрою: 0 для Slave та 1 для Master пристрою.

Регістр керування SPICR (рис. 6.43) задає режим *Master/Slave*, частоту послідовного обміну, фазу та полярність імпульсів, дозвіл переривання.



Рисунок 6.43. Формат регістра керування SPI SPICR

На рис. 6.43 позначено:

Біт **SPIE** (*Serial Peripheral Interrupt Enable*) – дозвіл переривання SPI;

Біт **SPE** (*Serial Peripheral Output Enable*) – дозвіл виходу SPI;

Біт **SPR2** (*Divider Enable*) – дозвіл ділення частоти (табл. 6.28);

Біт **MSTR** (*Master Mode*) – режим *Master*;

Біт **CPOL** (*Clock Polarity*) - полярність імпульсів;

Біт **CPHA** (*Clock Phase*) – фаза імпульсів;

Біти **SPR[1:0]** (*Serial Clock Frequency*) – завдання частоти згідно табл. 6.28.

Таблиця 6.28 – Завдання частоти роботи SPI

Serial Clock	SPR2	SPR1	SPR0
$f_{CPU}/4$	1	0	0
$f_{CPU}/8$	0	0	0
$f_{CPU}/16$	0	0	1
$f_{CPU}/32$	1	1	0
$f_{CPU}/64$	0	1	0
$f_{CPU}/128$	0	1	1

Діаграми обміну за інтерфейсом SPI (рис. 6.44) пояснюють вибір полярності та фази тактових імпульсів.

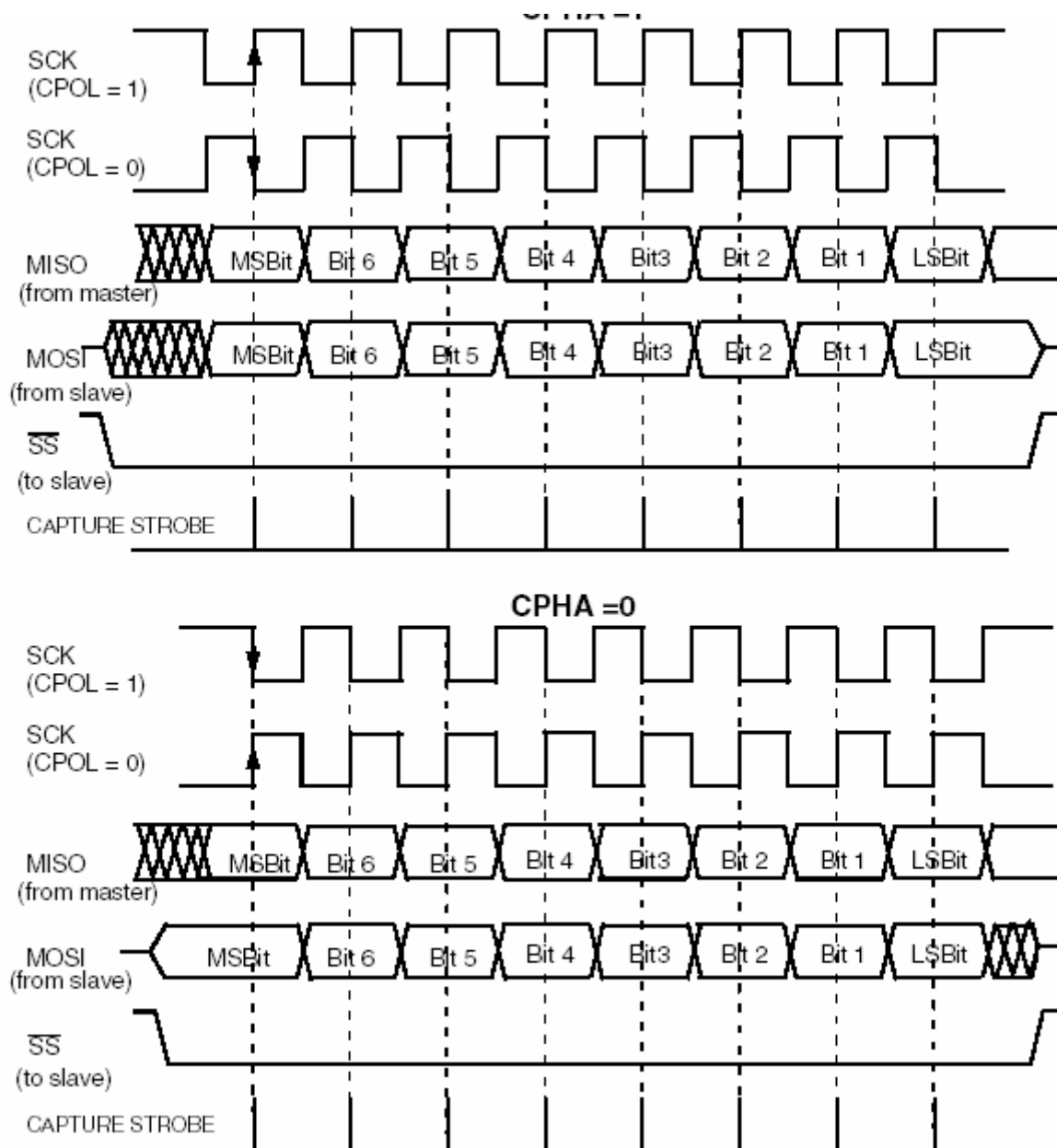


Рисунок 6.44 – Діаграми обміну за інтерфейсом SPI

Регістр керування/статусу SPICSR (рис. 6.45) містить наступні біти:



Рисунок 6.45. Формат регістра керування/ статусу SPICSR

Біт **SPIF** (*Serial Peripheral Data Transfer Flag*) – прапорець завершення передачі. Встановлюється апаратно. Значення **SPIF=0** свідчить про те, що триває процес передачі даних, або прапор був очищений. Значення **SPIF=1** свідчить про завершення передачі даних, при цьому запис у регістр **SPIDR** заборонений, поки відбувається читання регістру **SPICSR**;

Біт **WCOL** (*Write Collision status*) – помилка запису – завантаження регістру даних SPI під час передавання. Значення **WCOL=0** означає відсутність, а **WCOL=1** – наявність помилок запису;

Біт **OVR** (*SPI Overrun error*) – помилка переповнення. Встановлюється апаратно, коли дані готові до передачі у регістр **SPIDR**, а значення біту **SPIF**=1. Значення **OVR**=0 вказує на відсутність, а **OVR**=1 – наявність помилок переповнення;

Біт **MODF** (*Mode Fault flag*) – прапорець аварійного режиму. Значення **MODF**=0 вказує на відсутність, а **MODF**=1 – наявність помилок у роботі Master-пристрою;

Біт **SOD** (*SPI Output Disable*) – вихід SPI заборонений. Біт встановлюється та скидається програмно. Значення **SOD**=1 забороняє альтернативну функцію SPI-виходу (MOSI в режимі Master та MISO в режимі Slave). Значення **SOD**=0 вказує на дозвіл SPI-виходу, а **SOD**=1 – на його заборону;

Біт **SSM** (*SS Management*) – прапорець керування вибором Slave пристрою. Встановлюється та скидається програмно. Значення **SSM**=0 відповідає апаратному керуванню (від зовнішнього пристрою). Значення **SSM**=1 відповідає програмному керуванню за внутрішнім \overline{SS} – сигналом (зовнішній вивід SPI \overline{SS} при цьому вільний);

Біт **SSI** (*SS Internal Mode*) – при 0 вибір пристрою Slave. Встановлюється та скидається програмно. Працює як сигнал Chip Select, контролюючи рівень сигналу вибору пристрою Slave, коли **SSM**=1. Значення **SSI**=0 відповідає обраному пристрою Slave.

Приклад:

Ініціалізація регістру SPISR

SPIF	WCOL	OV	MODF	0	SOD	SSM	SSI	
0	0	0	0	0	0	1	1	\$03

```
1d A,#$03 ; завантаження константи $03 до регістру "A"
1d SPISR,A ; завантаження вмісту регістру "A" у регістр SPISR
```

Підпрограма ініціалізації інтерфейсу SPI

```
Init_SPI:
1d A,#$03 ; завантаження константи $03 до регістру "A"
; Біти 0 та 1
; регістру SPISR обрані такими SSM=1 та SSI = 1
; (вибір режиму Master)
1d SPISR,A ; завантаження вмісту регістру "A" у регістр SPISR
; SPI SW згенерує сигнал SS'
1d A,$5C ; ініціалізація порту SPI $5C = %01011100 (біт 0 ;
1 та 5)
; обираємо швидкість = 000
; SPR0=SPR1=SPR5=0 Frq = fcpu/8 (регістр SPICR)
; та біт 2 у регістрі SPICR = 1
; по другому фронту імпульсів тактового генератора
; та біт 3 = 1 регістру SPICR режим pull up
; SCK та bit4 =1 регістру SPICR
; (режим Master) та біт 6 =1 означає SPI під'єднано
; до зовнішнього пристрою
1d SPICR,A ; завантаження вмісту регістру "A" у регістр SPICR
ret
```

Окрім цього необхідно відповідним чином налаштувати обрані для SPI порти. А саме, для сигналу SPI SCK повинен бути обраний режим виводу Push Pull.

Аналого-цифровий перетворювач

Аналого-цифровий перетворювач являє собою семиканальний 10-розрядний АЦП послідовного наближення. Структурна схема АЦП наведена на рис.6.46.

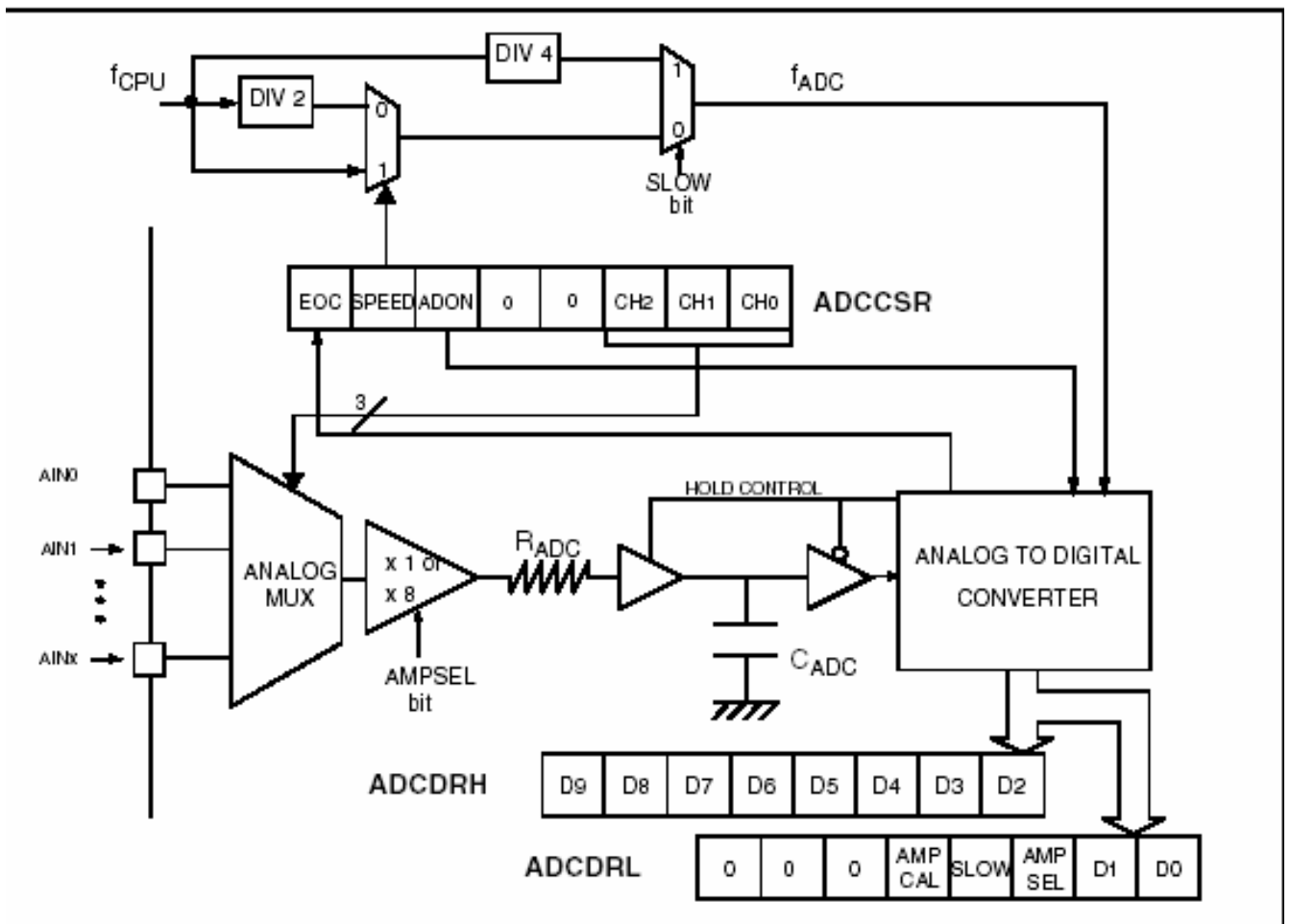


Рисунок 6.46 - Структурна схема АЦП

Схема містить: аналогово-цифровий перетворювач, аналоговий мультиплексор, вхідний підсилювач з регульованим коефіцієнтом підсилення (x1 або x8) та регістри даних ADCDRH, ADCDRL та керування/статусу ADCCSR. Регістр ADCDRH зберігає старші біти результату аналогового перетворення D[9:2], регістр ADCDRL – два молодших біти D[1:0] та крім того, наступні керуючі біти:

AMPCAL (Amplifier Calibration Bit) – калібровка підсилювача. При AMPCAL=1 ввімкнено режим калібровки, при цьому вхідна напруга підсилювача встановлюється на нульовому рівні;

SLOW – біт режиму Slow;

AMPSEL (Amplifier Selection Bit) – біт вибірки підсилювача (вибору коефіцієнту підсилення 1 або 8, див. рис. 6.46). При AMPSEL=1 діапазон вхідної напруги становить від 0 до $U_{CC}/8$ (при $U_{CC}=5\text{ В}$ – від 0 до 430 мВ). При цьому роздільна здатність дорівнює 0,6 мВ (еквівалент 13-го розряду). При AMPSEL=0 діапазон вхідної напруги становить від 0 до U_{CC} .

Регістр керування/статусу ADCCSR (див. рис. 6.46) містить наступні біти:

EOC (*End of Conversion*) – встановлюється в 1 по закінченню аналогово-цифрового перетворення;

SPEED (*ADC clock selection*) – біт, який сумісно з бітом SLOW регістра ADCDRL задає частоту роботи АЦП згідно табл.6.29;

ADON (*A/D Converter ON*) – запуск АЦП

Біти CH[2:0] (*Channel Selection*) – задають канал АЦП згідно з табл. 6.30.

Таблиця 6.29 – Вибір частоти АЦП перетворення

f_{ADC}	SLOW	SPEED
-----------	------	-------

$f_{CPU}/2$	0	0
f_{CPU}	0	1
$f_{CPU}/4$	1	x

Таблиця 6.30 – Вибір каналу АЦП

Вивід* МК	CH2	CH1	CH0
AIN0	0	0	0
AIN1	0	0	1
AIN2	0	1	0
AIN3	0	1	1
AIN4	1	0	0
AIN5	1	0	1
AIN6	1	1	0

*Примітка: номер виводу МК – див. табл. 6.15

Приклад 6. Написати підпрограму ініціалізації АЦП для введення аналогового сигналу по лінії PB0.

Для виконання цієї задачі необхідно лінію порту PB0 запрограмувати для введення у високоімпедансний стан (00) та обрати канал АЦП (000):

Init_ADC:

```

call    init_portB           ; ініціалізація порту B
ld      A, #00000000        ; Вибір AIN0 (PB0) скиданням CH0=CH1=CH2
ld      ADCCSR, A
ret

```

Приклад 6.* Введення даних з АЦП (8-розрядного)

Read_ADC:

```

bset    ADCCSR, #5          ; Запуск АЦП
cont:                                       ; Очікування закінчення перетворення

ld      A, ADCCSR
and     A, #80
jreq    cont
bres    ADCCSR, #5          ; Зупинка АЦП встановленням біта 5 (ADON)
                                       ; регістра ADCCSR
ld      A, ADCDRH           ; Зчитування молодшого байту результату АЦП
ld      var, A              ; запис у комірку var
ret

```

Таймери

Сторожовий таймер (WDG) використовується для виявлення випадку програмної помилки або несподіваної логічної ситуації, яка приводить програмний додаток до відхилення від нормального виконання програми. Сторожовий таймер генерує скидання МК після закінчення заданого періоду часу. Таймер (рис. 6.47) містить програмовний автономний лічильник зворотної дії та реєстр керування WDGCR.

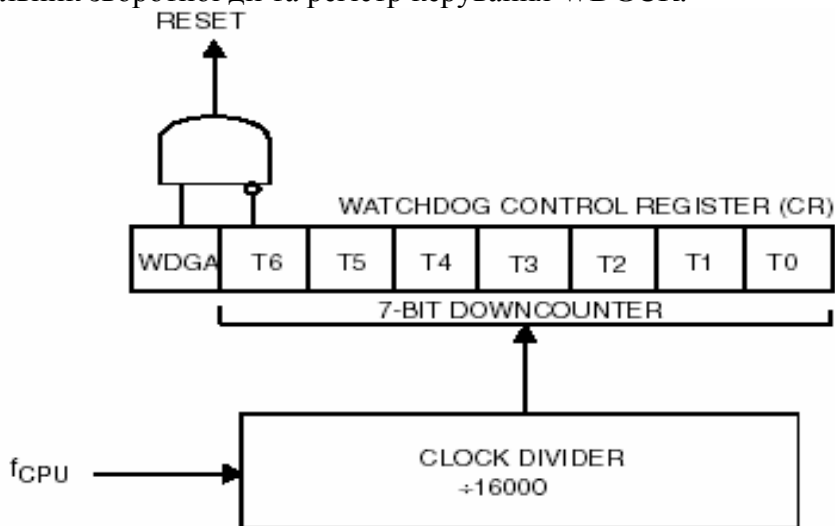


Рисунок 6.47 – Функціональна схема сторожового таймеру (WDG) та реєстру керування

Значення лічильника декрементується кожні 16000 машинних циклів. Біт WDGA реєстра WDGCR (біт активації) встановлюється програмно, а скидається тільки апаратно після скидання. Якщо WDGA=0, то WDG відключений, якщо WDGA=1, то WDG активний і може генерувати скидання.

Біти T[6:0] містять значення, яке буде декрементуватися. Скидання відбудеться, коли значення зміниться від 40h до 3Fh (біт T6 стане рівним 0). Після скидання таймер не активований. Значення реєстру керування WDGCR при скиданні: 0111 1111 (7Fh). Після активації таймеру (WDGA=1) значення бітів T[6:0] змінюються, і при досягненні T6=0 формується сигнал RESET (див. рис. 6.47). Програмний додаток повинен записувати в WDGCR реєстр значення в певні інтервали під час нормальної роботи, щоб запобігти скиданню МК.

Порядок включення сторожового таймера наступний:

- встановлення біту WDGA=1 (активація сторожового таймеру);
- встановлення біту T6=1 для запобігання генерації миттєвого скидання:


```
LD A,#$FF
LD WDGCR, A
```
- встановлення бітів T[5:0], які містять кількість операцій декрементування, що визначає часову затримку перед генерацією скидання.

Таблиця 6.31. Час затримки сторожового таймера

$f_{CPU} = 8 \text{ МГц}$		
Значення реєстру CR	Мінімальний час затримки, мс	Максимальний час затримки, мс
C0h	1	2
C1h	2	3
...
FFh	127	128

Зазначимо, що якщо сторожовий таймер активний, то команда HALT згенерує скидання. Після активації сторожовий таймер не може бути вимкнений, крім випадку скидання.

Таймер ART являє собою 12-розрядний таймер з автоперезавантаженням, призначений для:

- реалізації часових затримок;
- генерації переривань при переповненні, захопленні, порівнянні;
- генерації 4 незалежних ШІМ-сигналів.

Таймер побудований на автономному (несинхронізованому) 12-розрядному інкрементному лічильнику з вхідним автоперезавантажувальним регістром і 4-ма вихідними PWM (Pulse-Width Modulation – ШІМ) каналами. Таймер має 6 зовнішніх виводів:

- 4 виходи ШІМ;
- вхід АТІС (AT Input Capture) для реалізації функції захоплення введення;
- вхід BREAK для ШІМ (для припинення сигналу на PWM виводах).

Основні характеристики ART наступні:

- частота рахування – 2 кГц – 4 МГц при $f_{CPU} = 8$ МГц;
- керування полярністю сигналів на входах і виходах;
- масковані переривання при переповненні таймеру, порівнянні та захопленні.

Реалізації часових затримок. При переповненні 12-розрядного лічильника CNTR встановлюється прапор переповнення OVF в регістрі контролю/статусу ARTCSR, що свідчить про закінчення певного інтервалу часу. Цей інтервал можна змінювати записом коду в регістр автоперезавантаження ATR та зміною частоти лічильних імпульсів. За одиничним значенням прапора переповнення OVF лічильник переходить з стану FFFh в стан коду ATR (значення автоперезавантаження).

Захоплення події. За переднім або заднім фронтом на виводі АТІС мікроконтролера вміст 12-розрядного лічильника CNTR запам'ятовується у регістрі АТІСР; при цьому встановлюється біт ICF; якщо переривання дозволено (біт ІСІЕ=1), воно генерується. Біт ICF скидається при читанні АТІСР регістра. Регістр АТІСР доступний лише для читання і завжди містить значення інкрементуючого лічильника, яке відповідає останньому захопленню вхідних даних. Будь-яке подальше захоплення даних забороняється, поки біт ICF=1.

Режим порівняння. Щоб використовувати цю функцію, необхідно завантажити 12-розрядне значення в регістри DCRxH і DCRxL. Коли лічильник CNTR досягне значення, яке зберігається в DCRxH і DCRxL регістрах, то біт СМРF в регістрі PWMxCSR встановиться в 1 і згенерує запит на переривання, якщо встановлений біт дозволу переривання СМРІЕ=1.

Генерація ШІМ сигналів. PWM режим дозволяє згенерувати до 4-х ШІМ сигналів. PWMx вихідні сигнали можуть бути дозволені або заборонені бітом ОЕх в регістрі PWMCR. Чотири PWM сигнали мають однакову частоту (f_{PWM}), яка задається частотою лічильника і значенням регістра ATR:

$$f_{PWM} = f_{COUNTER} / (4096 - ATR) . \quad (6.1)$$

З формули (6.1) витікає наступне:

- якщо $f_{COUNTER} = 32$ МГц, то максимальне значення $f_{PWM} = 8$ МГц (значення регістру ATR = 4092), мінімальне значення $f_{PWM} = 8$ КГц (значення регістру ATR = 0);
- якщо $f_{COUNTER} = 4$ МГц, то максимальне значення $f_{PWM} = 2$ МГц (значення регістру ATR = 4094), мінімальне значення $f_{PWM} = 1$ КГц (значення регістру ATR = 0).

Функція останову (*Break*) активується зовнішнім сигналом BREAK (рис. 6.48) на однойменному виводі (активний рівень - 0). BREAK вивід повинен бути заздалегідь дозволений програмно шляхом встановлення BPEN біта у реєстрі BREAKCR. Коли на BREAK виводі присутній сигнал низького рівня, біт BA встановлюється в 1, і функція зупинки активується. Послідовність всіх 4 ШІМ припиняється, 12-бітний PWM лічильник, реєстри ARR, PWMCR, DCRx і відповідні тіньові реєстри встановлюються в свої початкові значення при скиданні.

Рисунок 6.48 – Дія сигналу BREAK на ШІМ виходи

Опис реєстрів таймеру.

Реєстр керування/стану таймера ATCSR (AUTORELOAD TIMER CONTROL STATUS REGISTER) допускає читання та запис. Значення при скиданні: 0x00 0000. Формат реєстра ATCSR подано на рис.6.49 .

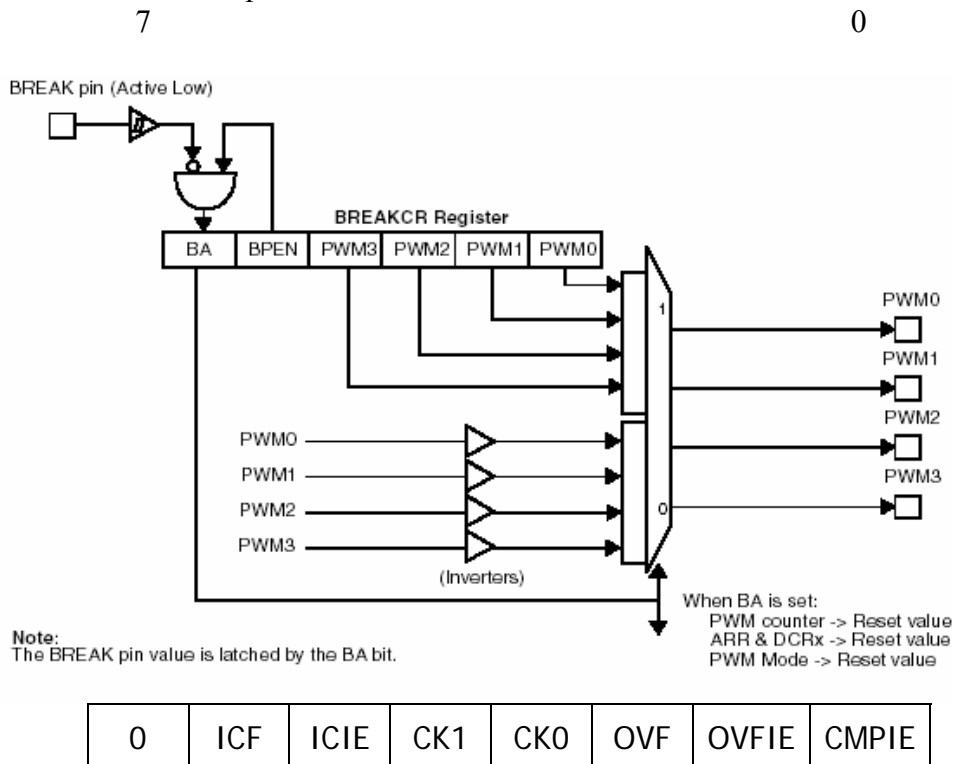


Рисунок 6.49 – формат реєстра ATCSR

На рис. 6.49 позначено:

Біт ICF (*Input Capture Flag*) – прапорець вхідного захоплення. Цей біт встановлюється апаратно і скидається програмно при читанні ATICR реєстра (дозвіл на читання з ATICRH або ATICRL скидатиме цей прапорець). Запис в цей біт не змінює його значення. Якщо ICF=1, то вхідне захоплення виконано.

Біт ICIE (*IC Interrupt Enable*) – дозвіл IC переривання. Цей біт встановлюється і скидається програмно. Якщо біт дорівнює 0, то переривання вхідного захоплення заборонено, якщо 1 – переривання вхідного захоплення дозволено.

Біти CK[1:0] (*Counter Clock Selection*) визначають вибір генератора лічильника. Ці біти встановлюються і скидаються програмно і скидаються апаратно після сигналу RESET. Вони визначають частоту генератора лічильника (табл.6.32). Зміна вступає в дію після переповнення.

Біт OVF (*Overflow Flag*) – прапорець переповнення. Цей біт встановлюється апаратно і скидається програмно при читанні TCSR реєстра. Він відображує перехід стану лічильника

з FFFh в ATR значення (автоперезавантаження). Якщо OVF=0, то переповнення лічильника не відбулося, якщо OVF=1 –відбулося.

Таблиця 6.32. Вибір генератора лічильника

Вибір генератора лічильника	СК1	СК0
Вимк.	0	0
f_{TIMER} (період 1мс при $f_{\text{cpu}}=8$ МГц)	0	1
f_{cpu}	1	0
32 МГц	1	1

Біт OVFIE (*Overflow Interrupt Enable*) – дозвіл переривання від переповнення. Цей біт встановлюється/скидається програмно і скидається апаратно після сигналу RESET. Якщо OVFIE=0, то OVF переривання заборонено, якщо OVFIE=1, то OVF переривання дозволене.

Біт CMPIE (*Compare Interrupt Enable*) – дозвіл переривання захоплення. Цей біт встановлюється/скидається програмно і скидається апаратно після сигналу RESET. Якщо CMPIE=0, то переривання захоплення заборонено, якщо CMPIE=1, то переривання захоплення дозволено.

Старший регістр лічильника CNTRH (COUNTER REGISTER HIGH) допускає тільки читання. Значення при скиданні: 0000 0000.

Формат регістру CNTRH подано на рис.6.50.

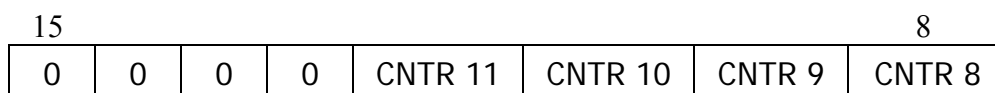


Рисунок 6.50 – формат регістра CNTRH

Молодший регістр лічильника CNTRL (COUNTER REGISTER LOW) допускає тільки читання. Значення при скиданні: 0000 0000. Формат регістру CNTRL подано на рис.6.51.

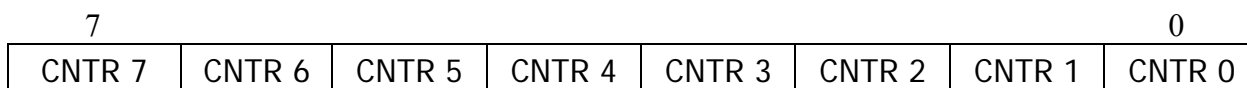


Рисунок 6.51 – формат регістра CNTRL

Біти CNTR [11:0] (*Counter Value*) – значення лічильника. Лічильник інкрементується з кожним тактовим імпульсом, як тільки генератор лічильника вибраний (див. табл. 6.32). Для зчитування значення лічильника використовують дві послідовні операції читання. При переповненні лічильника в нього заноситься значення ATR регістра.

Автоперезавантажувальний регістр ATRH (AUTORELOAD REGISTER HIGH) допускає читання та запис. Значення при скиданні: 0000 0000. Формат регістру ATRH подано на рис.6.52.

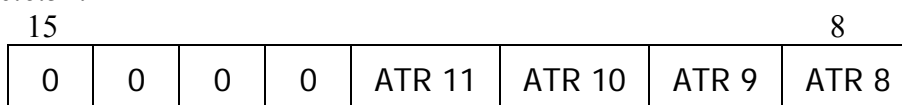


Рисунок 6.52 – формат регістра ATRH

Автоперезавантажувальний регістр ATRL (*AUTORELOAD REGISTER LOW*) допускає читання та запис. Значення при скиданні: 0000 0000. Формат регістру ATRL подано на рис.6.53.

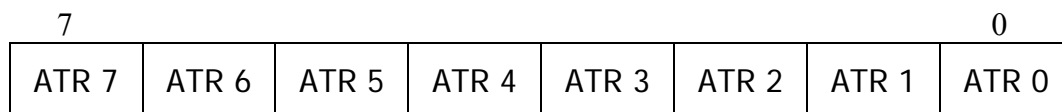


Рисунок 6.53 – формат регістра ATRL

Біти ATR [11:0] (*Autoreload Register*) – значення автоперезавантажувального регістру. Значення ATR автоматично завантажується в інкрементуючий лічильник CNTR при його переповненні OVF. В режимі ШІМ значення регістра ATR використовується для завдання частоти вихідного сигналу з ШІМ згідно виразу (6.1).

Регістр керування виводом PWM – PWMCR (PWM OUTPUT CONTROL REGISTER) допускає читання та запис. Значення при скиданні: 0000 0000. Формат регістру PWMCR подано на рис.6.54.

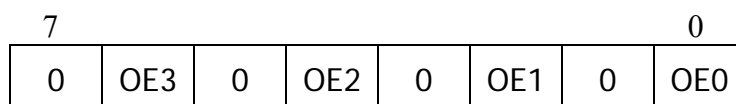


Рисунок 6.54 – формат регістра PWMCR

Біти OE [3:0] (*PWMx output enable*) – дозвіл виводу PWMx – встановлюються і скидаються програмно, а також скидаються апаратно після сигналу RESET. Якщо один з бітів дорівнює 0, то відповідний вивід PWM заборонено, якщо 1, то дозволено.

PWMx регістр керування/стану PWMx PWMxCSR (CONTROL STATUS REGISTER) допускає читання та запис. Значення при скиданні: 0000 0000. Формат регістру PWMxCSR подано на рис.6.55.

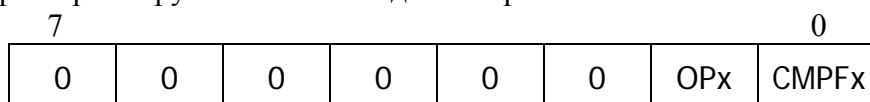


Рисунок 6.55 – формат регістра PWMxCSR

Біт OPx PWMx (*PWMx Output Polarity*) визначає полярність вихідного сигналу.

Біт CMPFx (*PWMx Compare Flag*). Цей біт встановлюється апаратно і скидається програмно при читанні PWMxCSR регістра. Він дорівнює 1, якщо значення інкрементуючого лічильника співпадає із значенням в регістрі DCRx.

Регістр керування зупинкою BREAKCR (BREAK CONTROL REGISTER) допускає читання та запис. Значення при скиданні: 0000 0000. Формат регістру BREAKCR подано на рис.6.56.

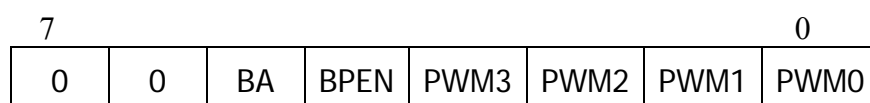


Рисунок 6.56 – формат регістра BREAKCR

На рис. 6.56 позначено:

Біт BA (*Break Active*) – зупинка дозволена. Цей біт встановлюється/скидається програмно, скидається апаратно після сигналу RESET і встановлюється апаратно, коли на виводі BREAK присутній нульовий рівень сигналу. Дію біта BA див. також на рис. 6.48.

Біт BPEN (*Break Pin Enable*) – дозвіл виводу BREAK. Цей біт встановлюється/скидається програмно і скидається апаратно після сигналу RESET. Якщо BPEN=0, то вивід BREAK заборонено, якщо BPEN= 1, то дозволено.

Біти PWM [3:0] (*Break Pattern*) – BREAK шаблон. Ці біти встановлюються/скидаються програмно і скидаються апаратно після сигналу RESET. Вони використовуються для переведення 4-х PWMx вихідних сигналів в стабільний стан, коли функція BREAK активна.

Старший PWMx регістр робочого циклу PWMx DCRxH (DUTY CYCLE REGISTER HIGH) допускає читання та запис. Значення при скиданні: 0000 0000. Формат регістру DCRxH подано на рис.6.57.

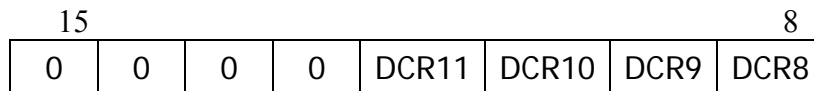


Рисунок 6.57 – формат регістра DCRxH

Молодший PWMx регістр робочого циклу PWMx DCRxL (DUTY CYCLE REGISTER LOW) допускає читання та запис. Значення при скиданні: 0000 0000. Формат регістру DCRxL подано на рис.6.58

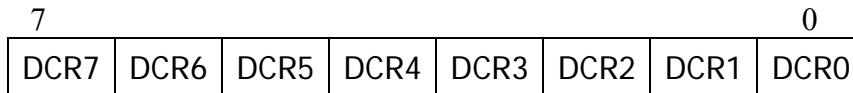
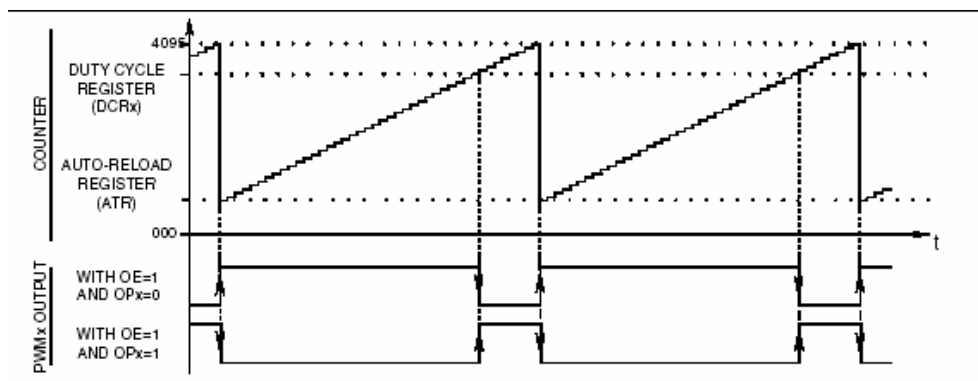


Рисунок 6.58 – формат регістра DCRxL

Біти DCR [11:0] (*PWMx Duty Cycle Value*) задають значення робочого циклу. У PWM режимі (OE_x=1 в регістрі PWMCR) біти DCR [11:0] визначають скважність вихідного сигналу ШІМ. У режимі порівняння вони визначають значення, яке порівнюватиметься з 12-розрядним значенням інкрементуючого лічильника. Формування імпульсів з ШІМ пояснюється рис. 6.59.

Рисунок 6.59 – формування ШІМ сигналів

Старший регістр вхідного захоплення ATICRH (INPUT CAPTURE REGISTER HIGH) допускає тільки читання. Значення при скиданні: 0000 0000. Формат регістру ATICRH подано на рис.6.60.



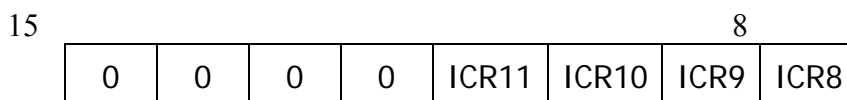


Рисунок 6.60 – формат регістра ATICRH

Молодший регістр вхідного захоплення ATICRL (INPUT CAPTURE REGISTER LOW) допускає тільки читання. Значення при скиданні: 0000 0000. Формат регістру ATICRL подано на рис.6.61.

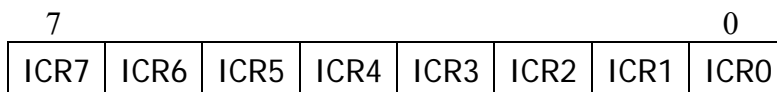


Рисунок 6.61 – формат регістра ATICRL

Біти ICR [11:0] (*Input Capture Data*) – дані вхідного захоплення. Регістр ATICR містить захоплене значення 12-розрядного регістра CNTR, тобто значення на момент появи переднього або заднього фронтів на виводі ATIC. Захоплення може бути виконано тільки тоді, коли прапорець ICF=0.

Регістр керування передачею TRANCR (TRANSFER CONTROL REGISTER) допускає читання та запис. Значення при скиданні: 0000 0001. Формат регістру TRANCR подано на рис.6.62.

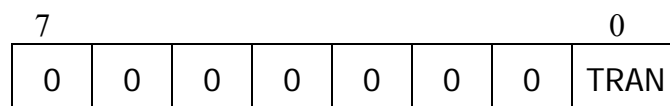


Рисунок 6.62 – формат регістра TRANCR

Біт TRAN (*Transfer enable*) - дозвіл перезапису DCRx у тінювий регістр після переповнення. Цей біт встановлюється/скидається програмно, скидається апаратно після кожної закінченої передачі і встановлюється апаратно після скидання RESET.

LITE - таймер використовується для функцій синхронізації. Таймер містить (рис.6.63) два 8-розрядних лічильника: лічильник 1 прямої дії з періодом 1 мс або 2 мс (при тактовій частоті 8 МГц) та лічильник 2 прямої дії з автоперезавантаженням та програмованим періодом від 4 мкс до 1,024 мс з інкрементом 4 мкс (при тактовій частоті 8 МГц), а також 8-розрядний регістр вхідного захоплення (LTICR).

Таймер генерує 2 маскованих переривання від переповнення лічильників та масковане переривання захоплення з можливістю виходу (пробудженням) з режиму Halt.

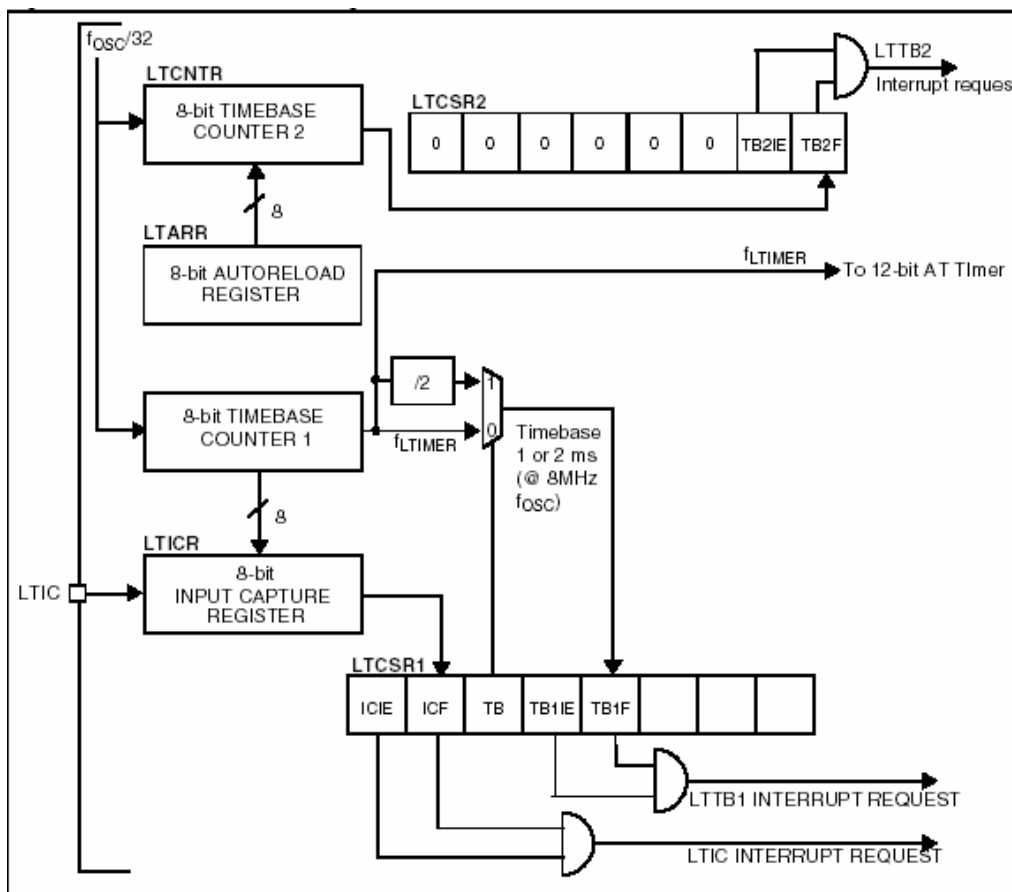


Рисунок 6.63 - структурна схема LITE-таймеру

Значення лічильника 1 не може бути прочитано або записано програмно. Після перезавантаження МК лічильник 1 починає відлік із значення 00 з частотою $f_{osc}/32$. Коли вміст лічильника досягає значення F9h, відбувається переповнення, після цього в лічильник записується значення 00. Якщо $f_{osc}=8$ МГц, то період між двома подіями переповненнями лічильника складає 1 мс. Цей період може бути подвоєний встановленням біта TB в регістрі LTCSR1.

При переповненні лічильника 1 апаратно встановлюється біт TB1F і, якщо біт TB1E=1 (переривання дозволено), генерується запит переривання. Біт TB1F скидається програмно при читанні регістра LTCSR1.

Лічильник 2 можна прочитати шляхом доступу до регістра LTCNTR. Після перезавантаження МК він інкрементується з частотою $f_{osc}/32$, починаючи зі значення, яке зберігається в регістрі LTARR. Коли вміст лічильника досягає значення FFh, відбувається переповнення, після якого в лічильник записується значення, завантажене в регістрі LTARR. Програмно можна записувати нове значення у будь-який час в регістр LTARR, це значення буде автоматично завантажено в лічильник, коли відбудеться чергове переповнення.

Коли лічильник 2 переповнюється, біт TB2F в регістрі LTCSR2 виставляється апаратно і генерується запит переривання, якщо біт TB2E=1 (переривання дозволено). Біт TB2F скидається програмно при читанні регістра LTCSR2.

Регістр вхідного захоплення (Input Capture) – 8-розрядний регістр, який використовується для захоплення незалежного інкрементуючого лічильника 1 після того, як буде знайдений передній або задній фронт на виводі LTIC. Коли відбудеться вхідне захоплення, то біт ICF буде встановлено в 1, а у регістр LTICR1 буде завантажений вміст лічильника 1. Переривання згенерується, якщо буде ICIE=1 (переривання дозволено). Біт ICF скидається при читанні регістра LTICR.

LTICR реєстр призначений тільки для читання і завжди містить дані з попереднього вхідного захоплення. Вхідне захоплення заборонено, якщо ICF=1.

Опис реєстрів таймера.

Реєстр 1 керування/стану LITE – таймера LTCSR1 (LITE TIMER CONTROL/STATUS REGISTER1) допускає читання та запис. Значення при скиданні: 0x00 0000. Формат реєстру наведено на рис.6.64.

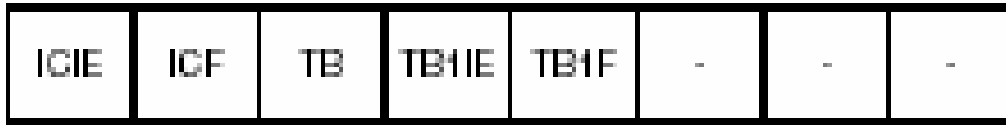


Рисунок 6.64 – Формат реєстра LTCSR1

Реєстр містить наступні біти:

Біт ICIE (*Interrupt Enable*) - дозвіл переривання вхідного захоплення. Цей біт встановлюється/скидається програмно. Якщо ICIE=0, то переривання вхідного захоплення заборонено, якщо ICIE=1, то дозволено.

Біт ICF (*Input Capture Flag*) – прапорець вхідного захоплення. Цей біт встановлюється апаратно та скидається програмно при читанні реєстру LTCSR1. Значення ICF=1 вказує, що вхідне захоплення відбулося.

Біт TB (*Timebase period selection*) – вибір періоду імпульсів лічильника 1. Цей біт встановлюється/скидається програмно. Якщо TB=0, період роботи лічильника 1 визначається як: $\text{Timebase period} = T_{\text{OSC}} * 8000$ (тобто складає 1 мс при тактовій частоті 8 МГц), при TB=1 - $\text{Timebase period} = T_{\text{OSC}} * 16000$ (2 мс при тактовій частоті 8 МГц).

Біт TB1IE (*Timebase Interrupt enable*) - дозвіл переривання лічильника 1. Цей біт встановлюється/скидається програмно. Одиночне значення біту вказує на дозвіл переривання від переповнення лічильника 1.

Біт TB1F (*Timebase Interrupt Flag*) - прапорець переривання лічильника 1. Цей біт встановлюється апаратно та скидається програмно при читанні реєстру LTCSR1. Одиночне значення біту вказує на те, що переривання від переповнення лічильника 1 відбулося.

Реєстр 2 керування/стану LITE – таймера LTCSR2 (LITE TIMER CONTROL/STATUS REGISTER2) допускає читання та запис. Значення при скиданні: 0000 1111. Формат реєстру подано на рис.6.65.

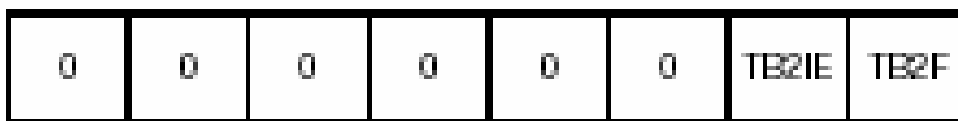


Рисунок 6.65 - Формат реєстра LTCSR2

Реєстр містить наступні біти:

Біт TB2IE (*Timebase 2 Interrupt enable*) – дозвіл переривання лічильника 2. Цей біт встановлюється/скидається програмно. Одиночне значення біту вказує на дозвіл переривання від переповнення лічильника 2.

Біт TB2F (*Timebase 2 Interrupt Flag*) – прапорець переривання лічильника 2. Цей біт встановлюється апаратно та скидається програмно при читанні реєстру LTCSR2. Одиночне значення біту вказує на те, що переривання від переповнення лічильника 2 відбулося.

Реєстр автоперезавантаження LITE – таймера LTARR (LITE TIMER AUTORELOAD REGISTER) допускає читання та запис. Значення при скиданні: 0000 0000. Формат реєстру подано на рис.6.66.



Рисунок 6.66 - Формат регістра LTARR

Регістр містить наступні біти:

Біти AR[7:0] (*Counter 2 Reload Value*) - значення кода перезавантаження лічильника 2. Цей код перезаписується із регістра LTARR у регістр LTCNTR2 при кожному переповненні лічильника 2.

Регістр лічильника 2 LITE – таймера LTCNTR (LITE TIMER COUNTER 2) допускає лише читання. Значення при скиданні: 0000 0000. Формат регістру подано на рис.6.67.



Рисунок 6.67 - Формат регістра LTCNTR

Регістр містить наступні біти:

Біти CNT[7:0] (*Counter 2 Reload Value*) – значення лічильника 2. Після переповнення в нього завантажуються вміст регістру LTARR .

Регістр вхідного захоплення LTICR (LITE TIMER INPUT CAPTURE REGISTER) допускає лише читання. Значення при скиданні: xxxx xxxx. Формат регістру наведено на рис.6.68.



Рисунок 6.68 - Формат регістра LTICR

Регістр містить наступні біти:

Біти ICR[7:0] (*Input Capture Value*) – дані вхідного захоплення. Регістр LTICR містить захоплене значення 8-розрядного лічильника 2, тобто значення на момент появи переднього або заднього фронту на виводі LTIC. Захоплення може бути виконано тільки тоді, коли прапорець ICF=0.

Типи адресації та систему команд МК ST7LITE2 наведено у Додатку Г.

Приклади:

Підпрограма переривання для “режиму порівняння”

```
.artim_oc_rt      ; підпрограма буде запущена коли лічильник досягне
значення,
                ;що записано в регістри DCR0H and DCR0L і, тоді,
                ;буде встановлено прапорець CMPF0 (прапорець порівня)

ld      A,PADR   ; переключити LED (що означає завантаження
                ;нового значення до порту A
xor     A,$01   ; виключаюче АБО з %00000001, що застосовується
                ;до вмісту регістру даних порту A
ld      PADR,A  ; завантаження результату розрахунків до порту
А

ld      A,{value0_5+1} ; додавання value0_5 до поточного вмісту
                ; регістру COMPARE
                ; (value0_5 це константа, яка дорівнює $01F4)
add     A,DCR0L ; додавання вмісту А до вмісту DCR0L (PWM0)
```

```

    1d      X,A      ; Завантаження вмісту A до регістру X
    1d      A,value0_5 ; Завантаження вмісту змінної value0_5 до
регістру A
    adc     A,DCR0H   ; Додавання з переносом вмісту A до
                    ; вмісту DCR0H (PWM0)
    1d      DCR0H,A   ; Запис результату до регістру DCR0H
    1d      DCR0L,X   ; завантаження вмісту X до регістру DCR0L з
PWM0
    1d      A,PWM0CSR ; Читаємо для очищення
                    ; прапорець SMPF0 (прапорець порівняння)
    ired   ; Повертаємо керування головній програмі після
                    ; завершення обробки цього переривання
                    ;Авто перезавантаження підпрограми таймера

```