



Для того, что бы программист всегда мог иметь "под рукой" часто ртебующуюся информацию, в комплект включена "Шпаргалка программиста".

Рекомендуется распечатать следующие 2 страницы на плотной бумаге и держать постоянно "под рукой".

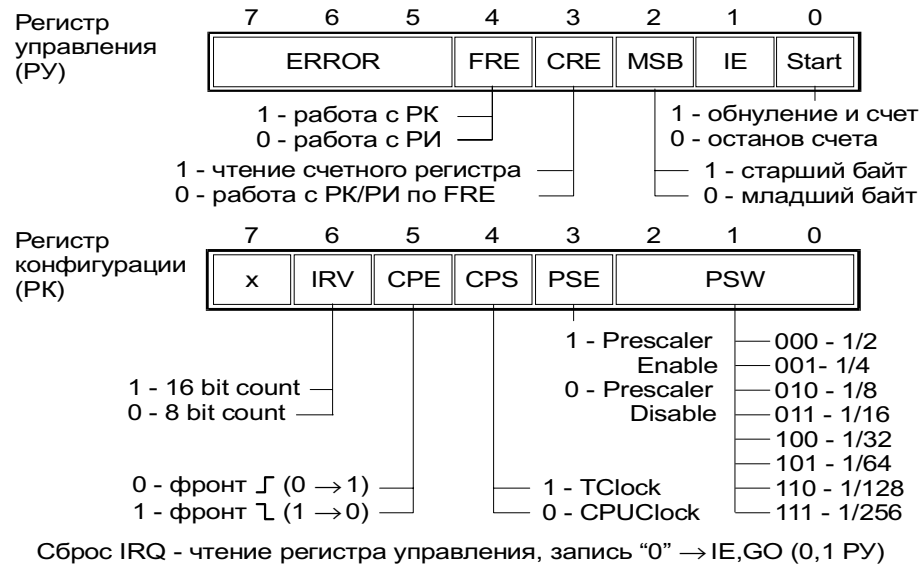
АНГСТРЕМ с благодарностью примет любые предложения, направленные как на улучшение описания микроконтроллера в целом, так и шпаргалки.

103460, **МОСКВА**, Зеленоград, ОАО **АНГСТРЕМ**
т. (095) 531-49-06, т/ф. 532-96-21
E-mail: market@angstrem.ru
<http://www.angstrem.ru>



Таймер

05h - адрес рабочего регистра
04h - адрес регистра управления



Параллельные порты A/B

Порт А
01h - адрес рабочего регистра
19h - адрес регистра конфигурации

Порт В
02h - адрес рабочего регистра
1Ah - адрес регистра конфигурации



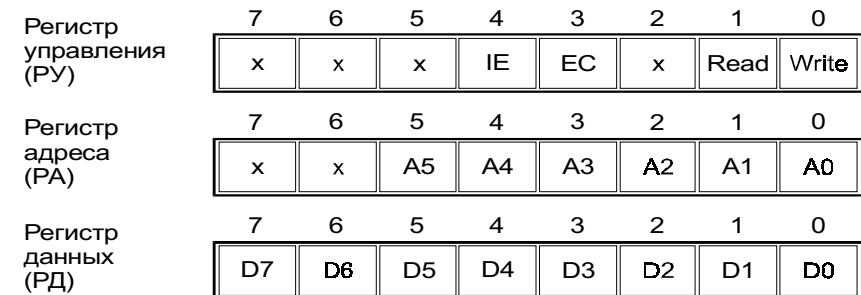
Регистр сторожевого таймера

1Dh - адрес регистра



Регистры ЭСППЗУ

38h - регистр управления
39h - регистр адреса
3Fh - регистр данных



Сброс IRQ - чтение/запись регистра управления

x - Разряд регистра не используется
IE - разрешение прерывания: 1 - разрешено, 0 - запрещено.
EC - Признак ошибки чтения: 1 - ошибка исправлена, 0 - нет реакции.



Наименование команды	Мнемоника	Код команды	Действие команды	Состояние S Z C of dC
Двухоперандные команды				
Пересылка	MOV	0000 01ss sssd dddd	src → dst	* * - 0 -
Сравнение	CMP	0000 10ss sssd dddd	dst - src S,Z,C → S	* * * * *
Сложение	ADD	0001 00ss sssd dddd	dst + src → dst	* * * * *
Вычитание	SUB	0000 11ss sssd dddd	dst - src → dst	* * * * *
Логическое И	AND	0001 01ss sssd dddd	dst .AND. src → dst	* * 0 0 0
Логическое ИЛИ	OR	0001 10ss sssd dddd	dst .OR. src → dst	* * 0 0 0
Исключающее ИЛИ	XOR	0001 11ss sssd dddd	dst .XOR. src → dst	* * 0 0 0
Литерные команды				
Пересылка литеры	MOVL	010c cccc cccd dddd	const → dst	* * - 0 -
Сравнение с литерой	CMPL	011c cccc cccd dddd	dst - const S,Z,C → RS	* * * * *
Сложение с литерой	ADDL	0011 00cc cccd dddd	dst + sconst → dst	* * * * *
Вычитание литеры	SUBL	0010 11cc cccd dddd	dst - sconst → dst	* * * * *
Сброс разрядов	BIC	0010 10pc cccd dddd	NOT(const).AND.dst → dst	* * 0 0 0
Установка разрядов	BIS	0011 10pc cccd dddd	dst .OR. tconst → dst	* * 0 0 0
Инверсия разрядов	BTG	0011 11pc cccd dddd	dst.XOR. tconst → dst	* * 0 0 0
Проверка разрядов	BTT	0011 01pc cccd dddd	dst.AND. tconst, S,Z → RS	* * 0 0 0
Однооперандные команды				
Обмен тетрад	SWAP	0000 0000 001d dddd	dst(n) → dst(n+4) n<4 dst(n) → dst(n-4)	* * 0 0 0
Смена знака	NEG	0000 0000 010d dddd	-dst → dst	* * * * *
Инверсия всех разрядов	NOT	0000 0000 011d dddd	NOT(dst) → dst	* * - 0 -
Логический сдвиг влево	SHL	0000 0000 100d dddd	dst(n) → dst(n+1), 0 → dst(0), dst(7) → dst(7), dst(0) → C	* * * * 0
Логический сдвиг вправо	SHR	0000 0000 101d dddd	dst(n+1) → dst(n), 0 → dst(7), dst(0) → C	0 * * 0 0
Арифметический сдвиг вправо	SHRA	0000 0000 110d dddd	dst(n+1) → dst(n), dst(7) → dst(7), dst(0) → C	* * * 0 0
Циклический сдвиг влево	RLC	0000 0000 111d dddd	dst(n) → dst(n+1), C → dst(0), dst(7) → C	* * * * 0
Циклический сдвиг вправо	RRC	0000 0001 000d dddd	dst(n+1) → dst(n), C → dst(7), dst(0) → C	* * * 0 0
Сложение с переносом	ADC	0000 0001 001d dddd	dst + C → dst	* * * * *
Вычитание переноса	SBC	0000 0001 010d dddd	dst - C → dst	* * * * *

Наименование команды	Мнемоника	Код команды	Действие команды	Состояние S Z C of dC
Команды работы со служебными регистрами и регистром состояния				
Загрузка служебных регистров	LDR	0010 0ccc cccc cnnn	const → reg	- - - - -
Запись в служебные регистры	MTPR	0000 0010 nnnn ssss	src → reg	- - - - -
Чтение служебных регистров	MFPR	0000 0011 nnnn dddd	reg → dst	- - - - -
Запись в стек данных	PUSH	0000 0000 0001 0nnn	reg → data stack, DSP = DSP + 1	- - - - -
Чтение из стека данных	POP	0000 0000 0001 1nnn	data stack → reg, DSP = DSP - 1	- - - - -
Установка разрядов RS	SST	0000 0001 1000 bbbb	if mask(n) = 1 then RS(n) = 1	* * * - -
Сброс разрядов RS	CST	0000 0001 1100 bbbb	if mask(n) = 1 then RS(n) = 0	* * * - -
Проверка переполнения	TOF	0000 0000 0000 0100	OF → Z	- * - - -
Проверка тетрадного переноса	TDC	0000 0000 0000 0101	DC → Z	- * - - -
Команды передачи управления				
Безусловный переход	JMP	1000 00aa aaaa aaaa	address → PC	- - - - -
Переход к подпрограмме	JSR	1001 00aa aaaa aaaa	PC → istack, address → PC, ISP = ISP + 1	- - - - -
Переход по Z=0 (не равно)	JNZ (JNE)	1011 00aa aaaa aaaa	address → PC if Z = 0	- - - - -
Переход по Z=1 (равно)	JZ (JEQ)	1010 00aa aaaa aaaa	address → PC if Z = 1	- - - - -
Переход по S=0 (плюс)	JNS	1100 00aa aaaa aaaa	address → PC if S = 0	- - - - -
Переход по S=1 (минус)	JS	1101 00aa aaaa aaaa	address → PC if S = 1	- - - - -
Переход по C=0	JNC	1110 00aa aaaa aaaa	address → PC if C = 0	- - - - -
Переход по C=1	JC	1111 00aa aaaa aaaa	address → PC if C = 1	- - - - -
Косвенный переход	IJMP	0000 0000 0000 0011	IR1 → PC	- - - - -
Косвенный переход к подпрограмме	IJSR	0000 0000 0000 0111	PC → istack, IR1 → PC, ISP = ISP + 1	- - - - -
Возврат из подпрограммы	RTS	0000 0000 0000 1100	istack → PC, ISP = ISP - 1	- - - - -
Возврат из подпрограммы с битом C	RTSC	0000 0000 0000 111c	istack → PC, c → RS(0), ISP = ISP - 1	- - * - -
Возврат из прерывания	RTI	0000 0000 0000 1101	istack → PC, data stack → RS	* * * * *
Специальные команды				
Нет операции	NOP	0000 0000 0000 0000		- - - - -
Ожидание	WAIT	0000 0000 0000 0001	RS(3) = 1 (INT Enable)	- - - - -
Останов	STOP	0000 0000 0000 1000	RS(3) = 1 (INT Enable)	- - - - -
Сброс	RESET	0000 0000 0000 0010	DSP = 0, ISP = 0	- - - - -
Прогон стека команд	SKSP	0000 0000 0000 0110	ISP = ISP - 1	- - - - -