

Система команд
микроконтроллеров
С16х, ХС16х и ST10х

ООО "КАСКОД-ЭЛЕКТРО"

2005

Санкт-Петербург

Обозначения

Данная система команд является общей для микроконтроллеров серий SAx-C161x-xx, SAx-C164x-xx, SAx-8xC166-x, SAx-C167xx-xx и SAx-XC167xx фирмы Infineon и для микроконтроллеров серий ST10x167, ST10x168 и ST10x269 фирмы STMicroelectronics. В микроконтроллерах серий SAx-XC167xx и ST10x269 имеется целочисленный DSP-сопроцессор, для которого к общей системе команд добавлены дополнительные инструкции.

Условные обозначения

Rw_i	2-байтовый регистр общего назначения (POH) (R0, R1, R2, R3)
Rw, Rw_n, Rw_m	2-байтовый POH (R0, R1, ..., R15)
Rb, Rb_n, Rb_m	байтовый POH (RL0, RH0, ..., RL7, RH7)
reg	регистр специального назначения (SFR) или POH (если инструкция работает с типом данных BYTE и один из операндов - SFR, то доступ при адресации "reg" возможен только к младшему байту)
mem	прямая адресация в памяти слова или байта
[...]	косвенная адресация в памяти слова или байта (любой 2-байтовый POH может использоваться как косвенный указатель адреса, кроме арифметических, логических и инструкций сравнения, в которых разрешены только R0...R3)
bitaddr	указатель бита в битадресуемом пространстве памяти
bitoff	указатель SFR в битадресуемом пространстве памяти
#data _x	непосредственные данные (число значащих младших разрядов, которые используются в инструкции, представлены соответствующим добавлением "x")
#mask ₈	непосредственная 8-битовая маска, используемая для изменения нескольких разрядов
count	временная переменная, используемая в качестве счетчика
tmp	временная переменная для промежуточных вычислений

Обозначение круглыми скобками

op	непосредственные данные
(op)	содержимое op
(op _n)	содержимое разряда n операнда op
((op))	значение, взятое по адресу, равному содержимому op (т.е. содержимое op является указателем требуемого значения)

Инструкции умножения и деления

MDL, MDH	регистры источников и/или приемников для инструкций умножения и деления
MD	4-байтовый регистр, состоящий из регистров MDL и MDH и обращение может производиться только через MDL и MDH

Расширенные инструкции

Инструкции EXTxx изменяют стандартную схему адресации регистров (reg) для SFR/ESFR и страничную адресацию, использующую регистры DPPx.

#pag ₁₀	непосредственный 10-битовый номер страницы памяти
#seg ₈	непосредственный 8-битовый номер сегмента памяти
#irang ₂	2-битовое число (0...3), определяющее количество инструкций на которые распространяются условия изменения стандартной схемы адресации. Количество инструкций равно значению #irang ₂ , увеличенному на 1

Режимы адресации инструкций перехода

caddr	непосредственный 16-битовый внутрисегментный адрес перехода
seg ₈	непосредственный 8-битовый номер сегмента для перехода
rel	знаковое 8-битовое смещение по отношению к указателю инструкций IP
#trap ₇	непосредственный 7-битовый номер вектора прерывания

Коды условий перехода (cc)

	Условие	Номер	Описание
cc_UC	1 = 1	0h	безусловный переход
cc_Z	Z = 1	2h	если результат равен нулю
cc_NZ	Z = 0	3h	если результат не равен нулю
cc_V	V = 1	4h	если произошло переполнение во время выполнения инструкции
cc_NV	V = 0	5h	если не произошло переполнения во время выполнения инструкции
cc_N	N = 1	6h	если результат отрицательный

cc_NN	$N = 0$	7h	если результат неотрицательный
cc_C	$C = 1$	8h	если произошел перенос во время выполнения инструкции
cc_NC	$C = 0$	9h	если не произошел перенос во время выполнения инструкции
cc_EQ	$Z = 1$	2h	если сравниваемые операнды эквивалентны
cc_NE	$Z = 0$	3h	если сравниваемые операнды не эквивалентны
cc_ULT	$C = 1$	8h	меньше (без знака)
cc_ULE	$(Z \vee C) = 1$	Fh	меньше или равно (без знака)
cc_UGE	$C = 0$	9h	больше или равно (без знака)
cc_UGT	$(Z \vee C) = 0$	Eh	больше (без знака)
cc_SLT	$(N \oplus V) = 1$	Ch	меньше (со знаком)
cc_SLE	$(Z \vee (N \oplus V)) = 1$	Bh	меньше или равно (со знаком)
cc_SGE	$(N \oplus V) = 0$	Dh	больше или равно (со знаком)
cc_SGT	$(Z \vee (N \oplus V)) = 0$	Ah	больше (со знаком)
cc_NET	$(Z \vee E) = 0$	1h	если сравниваемые операнды не эквивалентны и один из операндов не равен наименьшему отрицательному числу

Мнемоническое обозначение операций

$(op1) \leftarrow (op2)$	$(op2)$ копируется в $(op1)$
$(op1) + (op2)$	$(op2)$ прибавляется к $(op1)$
$(op1) - (op2)$	$(op2)$ вычитается из $(op1)$
$(op1) * (op2)$	$(op2)$ умножается на $(op1)$
$(op1) / (op2)$	$(op1)$ делится на $(op2)$
$(op1) \wedge (op2)$	выполняется операция побитового логического И над операндами
$(op1) \vee (op2)$	выполняется операция побитового логического ИЛИ над операндами
$(op1) \oplus (op2)$	выполняется операция побитового исключающего ИЛИ над операндами

$(op1) \Leftrightarrow (op2)$ $(op2)$ сравнивается $(op1)$

$(op1) \bmod (op2)$ вычисляется остаток от деления $(op1)$ на $(op2)$

$\neg (op1)$ побитовая инверсия $(op1)$

Флаги состояния

*	стандартная установка значения флага, соответствующая функции флага и выполняемой инструкции
S	установка значения флага по правилам, отличающимся от стандартных
-	выполнение инструкции не влияет на флаг
0	значение флага всегда обнуляется при выполнении инструкции
NOR	флаг содержит инверсию результата выполнения операции логического ИЛИ над содержимым двух битовых операндов инструкции
AND	флаг содержит результат выполнения операции логического И над содержимым двух битовых операндов инструкции
OR	флаг содержит результат выполнения операции логического ИЛИ над содержимым двух битовых операндов инструкции
XOR	флаг содержит результат выполнения операции исключающего ИЛИ над содержимым двух битовых операндов инструкции
B	флаг содержит значение битового операнда до выполнения инструкции
B#	флаг содержит инверсию значения битового операнда до выполнения инструкции

Условные обозначения в мнемониках инструкций

....	каждый из четырех символов после “.” представляет собой один бит
..ii	2-битовый адрес ПОН (Rw_i : R0, R1, R2, R3)
..##	2-битовая непосредственная константа ($\#irang_2$)
:.###	3-битовая непосредственная константа ($\#data_3$)
c	4-битовый номер условия перехода cc
n	4-битовый номер ПОН (Rw_n или Rb_n)
m	4-битовый номер ПОН (Rw_m или Rb_m)
q	4-битовый номер разряда 2-байтового операнда QQ
z	4-битовый номер разряда 2-байтового операнда ZZ

#	4-битовая непосредственная константа (#data ₄)
t:ttt0	7-битовый непосредственный номер вектора прерывания
SS	8-битовый номер сегмента (0..255 для всех контроллеров, кроме C166; 2-битовый номер для C166, равный 0...3)
QQ	8-битовый адрес регистра POH или SFR (bitoff) для адресации бита
ZZ	8-битовый адрес регистра POH или SFR (bitoff) для адресации бита
rr	8-битовое смещение для относительных переходов (rel)
RR	8-битовый адрес регистра POH или SFR
##	8-битовая непосредственная константа (#data ₈)
## xx	8-битовая непосредственная константа, представленная двумя байтами, из которых содержимое байта xx не имеет значения
@@	8-битовая непосредственная константа (#mask ₈)
PP 0:00PP	10-битовый номер страницы
MM MM	16-битовый непосредственный внутрисегментный адрес (caddr) - младший и старший байт соответственно
## ##	16-битовая непосредственная константа (#data ₁₆) - младший и старший байт соответственно

Типы данных

Тип данных	Разрядность	Аналог на языке C/C++
BIT	1 бит	bool
BYTE	8 бит	signed char или unsigned char в зависимости от инструкции
WORD	16 бит	signed int или unsigned int в зависимости от инструкции
DOUBLEWORD	32 бита	signed long или unsigned long в зависимости от инструкции

Наименьшее отрицательное число для типа данных WORD равно минус 32768 (0x8000). Наименьшее отрицательное число для типа данных BYTE равно минус 128 (0x80).

Стек увеличивается в сторону младших адресов, поэтому при увеличении стека содержимое указателя стека SP уменьшается на 2 (т.к. стек работает только с данными типа WORD).

ADD Целочисленное сложение

Синтаксис ADD op1, op2

Действие $(op1) \leftarrow (op1) + (op2)$

Тип данных WORD

Описание Выполняется сложение содержимого двух операндов, представленных в дополнительном коде – источника op2 и приемника op1. Сумма сохраняется в op1.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
*	*	*	*	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Устанавливается, если произошло арифметическое переполнение, т. е. результат не может быть представлен указанным типом данных. В противном случае сбрасывается.
- C Устанавливается, если произошел перенос из старшего разряда для указанного типа данных. В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
ADD Rw_n, Rw_m	00 nm	2
ADD $Rw_n, [Rw_i]$	08 n:10ii	2
ADD $Rw_n, [Rw_i+]$	08 n:11ii	2
ADD $Rw_n, \#data_3$	08 n:0###	2
ADD $reg, \#data_{16}$	06 RR ## ##	4
ADD reg, mem	02 RR MM MM	4
ADD mem, reg	04 RR MM MM	4

ADDB Целочисленное сложение

Синтаксис ADDB op1, op2

Действие $(op1) \leftarrow (op1) + (op2)$

Тип данных BYTE

Описание Выполняется сложение содержимого двух операндов, представленных в дополнительном коде – источника op2 и приемника op1. Сумма сохраняется в op1.

Флаги состояния

E	Z	V	C	N
*	*	*	*	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Устанавливается, если произошло арифметическое переполнение, т. е. результат не может быть представлен указанным типом данных. В противном случае сбрасывается.
- C Устанавливается, если произошел перенос из старшего разряда для указанного типа данных. В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
ADDB Rb _n , Rb _m	01 nm	2
ADDB Rb _n , [Rw _i]	09 n:10ii	2
ADDB Rb _n , [Rw _i +]	09 n:11ii	2
ADDB Rb _n , #data ₃	09 n:0###	2
ADDB reg, #data ₈	07 RR ## xx	4
ADDB reg, mem	03 RR MM MM	4
ADDB mem, reg	05 RR MM MM	4

ADDC Целочисленное сложение с переносом

Синтаксис ADDC op1, op2

Действие $(op1) \leftarrow (op1) + (op2) + (C)$

Тип данных WORD

Описание Выполняется сложение содержимого трех операндов – источника op2, приемника op1 и бита переноса C, где операнды op1 и op2 – числа в дополнительном коде. Сумма сохраняется в op1. Эта инструкция может использоваться для вычислений с повышенной точностью.

Флаги состояния

E	Z	V	C	N
*	S	*	*	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю и флаг Z был установлен до выполнения инструкции. В противном случае сбрасывается.
- V Устанавливается, если произошло арифметическое переполнение, т. е. результат не может быть представлен указанным типом данных. В противном случае сбрасывается.
- C Устанавливается, если произошел перенос из старшего разряда. В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
ADDC Rw_n, Rw_m	10 nm	2
ADDC $Rw_n, [Rw_1]$	18 n:10ii	2
ADDC $Rw_n, [Rw_1+]$	18 n:11ii	2
ADDC $Rw_n, \#data_3$	18 n:0###	2
ADDC $reg, \#data_{16}$	16 RR ## ##	4
ADDC reg, mem	12 RR MM MM	4
ADDC mem, reg	14 RR MM MM	4

ADDCB Целочисленное сложение с переносом

Синтаксис ADDCB op1, op2

Действие $(op1) \leftarrow (op1) + (op2) + (C)$

Тип данных BYTE

Описание Выполняется сложение содержимого трех операндов – источника op2, приемника op1 и бита переноса C, где операнды op1 и op2 – числа в дополнительном коде. Сумма сохраняется в op1. Эта инструкция может использоваться для вычислений с повышенной точностью.

Флаги состояния

E	Z	V	C	N
*	S	*	*	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю и флаг Z был установлен до выполнения инструкции. В противном случае сбрасывается.
- V Устанавливается, если произошло арифметическое переполнение, т. е. результат не может быть представлен указанным типом данных. В противном случае сбрасывается.
- C Устанавливается, если произошел перенос из старшего разряда. В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
ADDCB Rb _n , Rb _m	11 nm	2
ADDCB Rb _n , [Rw ₁]	19 n:10ii	2
ADDCB Rb _n , [Rw ₁ +]	19 n:11ii	2
ADDCB Rb _n , #data ₃	19 n:0###	2
ADDCB reg, #data ₈	17 RR ## xx	4
ADDCB reg, mem	13 RR MM MM	4
ADDCB mem, reg	15 RR MM MM	4

AND Логическое И

Синтаксис AND op1, op2

Действие $(op1) \leftarrow (op1) \wedge (op2)$

Тип данных WORD

Описание Выполняется побитовая операция логического И над содержимым операндов источника op2 и приемника op1. Результат сохраняется в op1.

Флаги состояния

E	Z	V	C	N
*	*	0	0	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Всегда сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
AND Rw_n, Rw_m	60 nm	2
AND $Rw_n, [Rw_1]$	68 n:10ii	2
AND $Rw_n, [Rw_1+]$	68 n:11ii	2
AND $Rw_n, \#data_3$	68 n:0###	2
AND $reg, \#data_{16}$	66 RR ## ##	4
AND reg, mem	62 RR MM MM	4
AND mem, reg	64 RR MM MM	4

ANDB Логическое И

Синтаксис ANDB op1, op2

Действие $(op1) \leftarrow (op1) \wedge (op2)$

Тип данных BYTE

Описание Выполняется побитовая операция логического И над содержимым операндов источника op2 и приемника op1. Результат сохраняется в op1.

Флаги состояния

E	Z	V	C	N
*	*	0	0	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Всегда сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
ANDB Rb _n , Rb _m	61 nm	2
ANDB Rb _n , [Rw ₁]	69 n:10ii	2
ANDB Rb _n , [Rw ₁ +]	69 n:11ii	2
ANDB Rb _n , #data ₃	69 n:0###	2
ANDB reg, #data ₈	67 RR ## xx	4
ANDB reg, mem	63 RR MM MM	4
ANDB mem, reg	65 RR MM MM	4

ASHR Арифметический сдвиг в сторону младших адресов

Синтаксис ASHR op1, op2

Действие $(count) \leftarrow (op1)$
 $(V) \leftarrow 0, (C) \leftarrow 0$
 DO WHILE $((count) \neq 0)$
 $(V) \leftarrow (C) \vee (V), (C) \leftarrow (op1_0)$
 $(op1_n) \leftarrow (op1_{n+1}) [n = 0 \dots 14]$
 $(count) \leftarrow (count) - 1$
 END WHILE

Тип данных WORD

Описание Арифметически сдвигается содержимое приемника op1 на количество разрядов, определяемое источником op2. Сдвиг производится в сторону младших разрядов. Для сохранения знака операнда op1 старшие разряды результата заполняются нулями, если первоначально старший разряд был 0, или единицами, если старший разряд был 1. Флаг переполнения V используется как флаг округления. Младший значащий разряд сдвигается во флаг переноса C. Допустимые значения сдвига – от 0 до 15 включительно. Если op2 является регистром общего назначения, то используются только 4 младших бита.

Флаги состояния

E	Z	V	C	N
0	*	S	S	*

- E Всегда сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Устанавливается, если в любом цикле сдвига происходит переполнение (теряется 1 из флага C). Сбрасывается, если значение содержимого op2 равно 0.
- C Принимает значение бита, сдвигаемого из младшего разряда op1. Сбрасывается, если значение содержимого op2 равно 0.
- N Устанавливается, если установлен старший разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
ASHR Rw_n, Rw_m	AC nm	2
ASHR $Rw_n, \#data_4$	BC #n	2

АТОМІС Временное запрещение обработки запросов на прерывание

Синтаксис АТОМІС op1

Действие

```
(count) ← op1 [1 ≤ op1 ≤ 4]
IRQ_processing = Disable      // запрещение обработки
                               // запросов на
                               // прерывание
DO WHILE ((count) ≠ 0 AND Class_B_IRQ ≠ TRUE)
    ... // выполнение инструкции
    (count) ← (count) - 1
END WHILE
(count) = 0
IRQ_processing = Enable      // разрешение обработки
                              // запросов на
                              // прерывание
```

Описание Запрещается стандартная обработка запросов на прерывания от периферии и запросов класса А, а также обработка запросов от периферии контроллером PEC на время выполнения указанного количества инструкций. Действие АТОМІС распространяется уже на следующую инструкцию, поэтому не требуется дополнительных инструкций NOP. Количество инструкций, на которые распространяется действие АТОМІС, определяется операндом op1 и принимает значение от 1 до 4 вне зависимости от количества требуемых циклов шины. Если во время выполнения указанного количества инструкций пришел запрос класса В, действие АТОМІС прекращается и осуществляется обработка запроса в соответствии со стандартной процедурой.

Замечание Устройства SAX-8XC166-X не поддерживают АТОМІС.

Флаги состояния

E	Z	V	C	N
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
АТОМІС #irang ₂	D1:00##-0	2

BAND Битовое логическое И

Синтаксис BAND op1, op2

Действие $(op1) \leftarrow (op1) \wedge (op2)$

Тип данных BIT

Описание Выполняется операция логического И над содержимым источника op2 и содержимым приемника op1. Результат сохраняется в op1.

Флаги состояния

E	Z	V	C	N
0	NOR	OR	AND	XOR

- E Всегда сбрасывается.
- Z Результат выполнения операции ИЛИ-НЕ над содержимым операндов.
- V Результат выполнения операции ИЛИ над содержимым операндов.
- C Результат выполнения операции И над содержимым операндов.
- N Результат выполнения операции исключающее ИЛИ над содержимым операндов.

Формат инструкции

Мнемоника	Формат	Размер
BAND bitaddr _{z.z} , bitaddr _{o.o}	6A QQ ZZ qz	4

BCLR Очистка бита

Синтаксис BCLR op1

Действие (op1) ← 0

Тип данных BIT

Описание Обнуляется значение битового операнда op1. Эта инструкция обычно используется для управления системой и периферийными устройствами.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
0	B#	0	0	B

E Всегда сбрасывается.

Z Содержит инверсное значение предыдущего состояния указанного бита.

V Всегда сбрасывается.

C Всегда сбрасывается.

N Содержит предыдущее значение указанного бита.

Формат инструкции

Мнемоника	Формат	Размер
BCLR bitaddr _{q,q}	qE QQ	2

BCMP Сравнение битов

Синтаксис BCMP op1, op2

Действие (op1) \Leftrightarrow (op2)

Тип данных BIT

Описание Выполняется сравнение содержимого операнда op1 с содержимым операнда op2. Результат не сохраняется. Изменяются только флаги состояния.

Флаги состояния

E	Z	V	C	N
0	NOR	OR	AND	XOR

E Всегда сбрасывается.

Z Результат выполнения операции ИЛИ-НЕ над содержимым операндов.

V Результат выполнения операции ИЛИ над содержимым операндов.

C Результат выполнения операции И над содержимым операндов.

N Результат выполнения операции исключающего ИЛИ над содержимым операндов.

Формат инструкции

Мнемоника

BCMP bitaddr_{z.z}, bitaddr_{o.o}

Формат

2A QQ ZZ qz

Размер

4

BFLDH Битовое поле старшего байта

Синтаксис BFLDH op1, op2, op3

Действие $(tmp) \leftarrow (op1)$
 $(high\ byte(tmp)) \leftarrow ((high\ byte(tmp) \wedge \neg op2) \vee op3)$
 $(op1) \leftarrow (tmp)$

Тип данных WORD

Описание Производится побитная замена старшего байта содержимого приемника op1 значениями источника op3. Маской замены являются значения битов операнда op2. Если значение бита равно 1, то производится замена в соответствующем бите приемника.
 Те биты, которые необходимо оставить без изменения, должны иметь значение 0 в соответствующих разрядах op2 и op3. Иначе, если бит в op3 равен 1, то установится соответствующий бит в op1.

Флаги состояния

E	Z	V	C	N
0	*	0	0	*

- E Всегда сбрасывается.
- Z Устанавливается, если старший и младший байты результата равны нулю. В противном случае сбрасывается.
- V Всегда сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд старшего байта результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
BFLDH bitoff ₀ , #mask ₈ , #data ₈	1A QQ ## @@	4

BFLDL Битовое поле младшего байта

Синтаксис BFLDL op1, op2, op3

Действие $(tmp) \leftarrow (op1)$
 $(low\ byte(tmp)) \leftarrow ((low\ byte(tmp)) \wedge \neg op2) \vee op3$
 $(op1) \leftarrow (tmp)$

Тип данных WORD

Описание Производится побитная замена младшего байта содержимого приемника op1 значениями источника op3. Маской замены являются значения битов операнда op2. Если значение бита равно 1, то производится замена в соответствующем бите приемника.
 Те биты, которые необходимо оставить без изменения, должны иметь значение 0 в соответствующих разрядах op2 и op3. Иначе, если бит в op3 равен 1, то установится соответствующий бит в op1.

Флаги состояния

E	Z	V	C	N
0	*	0	0	*

- E Всегда сбрасывается.
- Z Устанавливается, если старший и младший байты результата op1 равны нулю. В противном случае сбрасывается.
- V Всегда сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд старшего байта результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
BFLDL bitoff ₀ , #mask ₈ , #data ₈	0A QQ @e ##	4

ВMOV Битовая пересылка

Синтаксис ВMOV op1, op2

Действие (op1) ← (op2)

Тип данных ВIT

Описание Содержимое источника op2 пересылается в приемник op1.
Флаги устанавливаются соответственно биту источника.

Флаги состояния

E	Z	V	C	N
0	B#	0	0	B

E Всегда сбрасывается.

Z Содержит инверсное значение предыдущего состояния источника.

V Всегда сбрасывается.

C Всегда сбрасывается.

N Содержит предыдущее состояние источника.

Формат инструкции

Мнемоника	Формат	Размер
ВMOV bitaddr _{z,z} , bitaddr _{q,q}	4A QQ ZZ qz	4

ВMOVN Битовая пересылка с инверсией

Синтаксис ВMOVN op1, op2

Действие $(op1) \leftarrow \neg(op2)$

Тип данных ВIT

Описание Инвертированное значение источника op2 пересылается в приемник op1. Флаги устанавливаются соответственно биту источника.

Флаги состояния

E	Z	V	C	N
0	B#	0	0	B

E Не изменяется.

Z Содержит инверсное значение предыдущего состояния источника.

V Не изменяется.

C Не изменяется.

N Содержит предыдущее состояние источника.

Формат инструкции

Мнемоника	Формат	Размер
ВMOVN bitaddr _{z,z} , bitaddr _{o,q}	3A QQ ZZ qz	4

BOR Битовое ИЛИ

Синтаксис BOR op1, op2

Действие $(op1) \leftarrow (op1) \vee (op2)$

Тип данных BIT

Описание Выполняется операция логического ИЛИ над содержимым источника op2 и приемника op1. Результат записывается в op1.

Флаги состояния

E	Z	V	C	N
0	NOR	OR	AND	XOR

E Всегда сбрасывается.

Z Результат выполнения операции ИЛИ-НЕ над содержимым операндов.

V Результат выполнения операции ИЛИ над содержимым операндов.

C Результат выполнения операции И над содержимым операндов.

N Результат выполнения операции исключающего ИЛИ над содержимым операндов.

Формат инструкции

Мнемоника

Формат

Размер

BOR bitaddr_{z.z}, bitaddr_{q.q}

5A QQ ZZ qz

4

BSET Установка бита

Синтаксис BSET op1

Действие (op1) ← 1

Тип данных BIT

Описание Содержимое операнда op1 устанавливается в 1. Эта инструкция обычно используется для управления системой и периферийными устройствами.

Флаги состояния

E	Z	V	C	N
0	B#	0	0	B

E Всегда сбрасывается.

Z Содержит инверсное значение предыдущего состояния указанного бита.

V Всегда сбрасывается.

C Всегда сбрасывается.

N Содержит предыдущее состояние указанного бита.

Формат инструкции

Мнемоника	Формат	Размер
BSET bitaddr _{q,q}	qF QQ	2

VXOR **Битовое исключающее ИЛИ**

Синтаксис VXOR op1, op2

Действие $(op1) \leftarrow (op1) \oplus (op2)$

Тип данных BIT

Описание Выполняется операция исключающего ИЛИ над содержимым источника op2 и приемника op1. Результат сохраняется в op1.

Флаги состояния

E	Z	V	C	N
0	NOR	OR	AND	XOR

E Всегда сбрасывается.

Z Результат выполнения операции ИЛИ-НЕ над содержимым операндов.

V Результат выполнения операции ИЛИ над содержимым операндов.

C Результат выполнения операции И над содержимым операндов.

N Результат выполнения операции исключающего ИЛИ над содержимым операндов.

Формат инструкции

Мнемоника

VXOR bitaddr_{z.z}, bitaddr_{o.o}

Формат

7A QQ ZZ qz

Размер

4

CALLA Абсолютный вызов подпрограммы

Синтаксис CALLA op1, op2

Действие

```

IF (op1) THEN
    (SP) ← (SP) - 2
    ((SP)) ← (IP)
    (IP) ← (op2)
ELSE
    ... // выполнение следующей инструкции
END IF

```

Описание Если выполняется условие, указанное в op1 (см. коды условий перехода), то осуществляется переход внутри сегмента по адресу, указанному в op2. Значение указателя стека SP уменьшается на 2 и содержимое указателя инструкций IP помещается на вершину системного стека. Поскольку IP всегда указывает на инструкцию, следующую за переходом, то значение, сохраненное в стеке, представляет собой адрес возврата для вызываемой подпрограммы. Если условие не выполняется, никаких действий не производится и следующая инструкция выполняется как обычно.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
CALLA cc, caddr	CA c0 MM MM	4

CALLI Косвенный вызов подпрограммы

Синтаксис CALLI op1, op2

Действие

```

IF (op1) THEN
    (SP) ← (SP) - 2
    ((SP)) ← (IP)
    (IP) ← (op2)
ELSE
    ... // выполнение следующей инструкции
END IF

```

Описание Если выполняется условие, указанное в op1 (см. коды условий перехода), то осуществляется переход внутри сегмента по адресу, равному содержимому op2. Значение указателя стека SP уменьшается на 2 и содержимое указателя инструкций IP помещается на вершину системного стека. Поскольку IP всегда указывает на инструкцию, следующую за переходом, то значение, сохраненное в стеке, представляет собой адрес возврата для вызываемой подпрограммы. Если условие не выполняется, никаких действий не производится и следующая инструкция выполняется как обычно.

Флаги состояния

E	Z	V	C	N
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
CALLI cc, [Rw _n]	AB cn	2

CALLR Относительный вызов подпрограммы

Синтаксис CALLR op1

Действие

$$(SP) \leftarrow (SP) - 2$$

$$((SP)) \leftarrow (IP)$$

$$(IP) \leftarrow (IP) + op1 * 2$$

Описание Осуществляется безусловный переход внутри сегмента по адресу, определяемому суммой указателя инструкций IP и удвоенного смещения op1. Смещение воспринимается как число в дополнительном коде. Значение указателя стека SP уменьшается на 2 и содержимое указателя инструкций IP помещается на вершину системного стека. Поскольку IP всегда указывает на инструкцию, следующую за переходом, то значение, сохраненное в стеке, представляет собой адрес возврата для вызываемой подпрограммы. Значение IP, используемое для вычисления адреса перехода, является адресом инструкции, следующей за CALLR.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
CALLR rel	BB rr	2

CALLS Межсегментный вызов подпрограммы

Синтаксис CALLS op1, op2

Действие

$$\begin{aligned} (SP) &\leftarrow (SP) - 2 \\ ((SP)) &\leftarrow (CSP) \\ (SP) &\leftarrow (SP) - 2 \\ ((SP)) &\leftarrow (IP) \\ (CSP) &\leftarrow op1 \\ (IP) &\leftarrow op2 \end{aligned}$$

Описание Осуществляется безусловный переход по полному 24-битовому адресу. Номер сегмента определяется операндом op1, а внутрисегментное смещение – операндом op2. Содержимое указателя команд IP и указателя сегмента CSP помещаются на вершину системного стека. Значение указателя стека SP перед каждым занесением на стек уменьшается на 2. Поскольку IP всегда указывает на инструкцию, следующую переходом, значение, сохраненное в стеке, представляет собой адрес возврата для вызываемой подпрограммы.

Флаги состояния

E	Z	V	C	N
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
CALLS seg, caddr	DA SS MM MM	4

CMP Целочисленное сравнение

Синтаксис CMP op1, op2

Действие (op1) \Leftrightarrow (op2)

Тип данных WORD

Описание Сравняется содержимое операнда op1 с содержимым операнда op2 посредством выполнения вычитания в двоичном дополнительном коде op2 из op1. Флаги устанавливаются по правилам вычитания. Операнды остаются неизменными.

Флаги состояния

E	Z	V	C	N
*	*	*	S	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу.
В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю.
В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен в указанном типе данных. В противном случае сбрасывается.
- C Устанавливается, если произошел заем.
В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
CMP Rw_n, Rw_m	40 nm	2
CMP $Rw_n, [Rw_i]$	48 n:10ii	2
CMP $Rw_n, [Rw_i+]$	48 n:11ii	2
CMP $Rw_n, \#data_3$	48 n:0###	2
CMP $reg, \#data_{16}$	46 RR ## ##	4
CMP reg, mem	42 RR MM MM	4

СМРВ Целочисленное сравнение

Синтаксис СМРВ op1, op2

Действие (op1) ⇔ (op2)

Тип данных BYTE

Описание Сравняется содержимое операнда op1 и операнда op2 посредством выполнения вычитания в двоичном дополнительном коде op2 из op1. Флаги устанавливаются по правилам вычитания. Операнды остаются неизменными.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
*	*	*	S	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу.
В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю.
В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен в указанном типе данных. В противном случае сбрасывается.
- C Устанавливается, если произошел заем.
В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
СМРВ Rb _n , Rb _m	41 nm	2
СМРВ Rb _n , [Rw _i]	49 n:10ii	2
СМРВ Rb _n , [Rw _i +]	49 n:11ii	2
СМРВ Rb _n , #data ₃	49 n:0###	2
СМРВ reg, #data ₁₆	47 RR ## xx	4
СМРВ reg, mem	43 RR MM MM	4

CMPD1 Целочисленное сравнение с уменьшением на единицу

Синтаксис CMPD1 op1, op2

Действие (op1) \Leftrightarrow (op2)
(op1) \leftarrow (op1) - 1

Тип данных WORD

Описание Содержимое приемника op1 сравнивается с содержимым источника op2 посредством выполнения вычитания в двоичном дополнительном коде op2 из op1. В качестве операнда op1 могут выступать только регистры РОН. После сравнения содержимое приемника op1 уменьшается на 1. Используя флаги и инструкции ветвления можно реализовывать любые циклы. Эта инструкция используется для уменьшения времени выполнения циклов.

Флаги состояния

E	Z	V	C	N
*	*	*	S	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен в указанном типе данных. В противном случае сбрасывается.
- C Устанавливается, если произошел заем. В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий бит результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
CMPD1 $Rw_n, \#data_4$	A0 #n	2
CMPD1 $Rw_n, \#data_{16}$	A6 Fn ## ##	4
CMPD1 Rw_n, mem	A2 Fn MM MM	4

CMPD2 Целочисленное сравнение с уменьшением на 2

Синтаксис CMPD2 op1, op2

Действие (op1) \Leftrightarrow (op2)
(op1) \leftarrow (op1) - 2

Тип данных WORD

Описание Содержимое приемника op1 сравнивается с содержимым источника op2 посредством выполнения вычитания в двоичном дополнительном коде op2 из op1. В качестве операнда op1 могут выступать только регистры РОН. После сравнения содержимое приемника op1 уменьшается на 2. Используя флаги и инструкции ветвления можно реализовывать любые циклы. Эта инструкция используется для уменьшения времени выполнения циклов.

Флаги состояния

E	Z	V	C	N
*	*	*	S	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен в указанном типе данных. В противном случае сбрасывается.
- C Устанавливается, если произошел заем. В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
CMPD2 R _{w_n} , #data ₄	B0 #n	2
CMPD2 R _{w_n} , #data ₁₆	B6 Fn ## ##	4
CMPD2 R _{w_n} , mem	B2 Fn MM MM	4

CMPI1 Целочисленное сравнение с увеличением на единицу

Синтаксис CMPI1 op1, op2

Действие (op1) \Leftrightarrow (op2)
(op1) \leftarrow (op1) + 1

Тип данных WORD

Описание Содержимое приемника op1 сравнивается с содержимым источника op2 посредством выполнения вычитания в двоичном дополнительном коде op2 из op1. В качестве операнда op1 могут выступать только регистры РОН. После сравнения содержимое приемника op1 увеличивается на 1. Используя флаги и инструкции ветвления можно реализовывать любые циклы.
Эта инструкция используется для уменьшения времени выполнения циклов.

Флаги состояния

E	Z	V	C	N
*	*	*	S	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен в указанном типе данных. В противном случае сбрасывается.
- C Устанавливается, если произошел заем. В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
CMPI1 $Rw_n, \#data_4$	80 #n	2
CMPI1 $Rw_n, \#data_{16}$	86 Fn ## ##	4
CMPI1 Rw_n, mem	82 Fn MM MM	4

CMPI2 Целочисленное сравнение с увеличением на 2

Синтаксис CMPI2 op1, op2

Действие (op1) \Leftrightarrow (op2)
(op1) \leftarrow (op1) + 2

Тип данных WORD

Описание Содержимое приемника op1 сравнивается с содержимым источника op2 посредством выполнения вычитания в двоичном дополнительном коде op2 из op1. В качестве операнда op1 могут выступать только регистры РОН. После всравнения содержимое приемника op1 увеличивается на 2. Используя флаги и инструкции ветвления можно реализовывать любые циклы.
Эта инструкция используется для уменьшения времени выполнения циклов.

Флаги состояния

E	Z	V	C	N
*	*	*	S	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен в указанном типе данных. В противном случае сбрасывается.
- C Устанавливается, если произошел заем. В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
CMPI2 R _{w_n} , #data ₄	90 #n	2
CMPI2 R _{w_n} , #data ₁₆	96 Fn ## ##	4
CMPI2 R _{w_n} , mem	92 Fn MM MM	4

CPL Поразрядная инверсия

Синтаксис CPL op1

Действие (op1) ← ¬(op1)

Тип данных WORD

Описание Вычисляется поразрядная инверсия содержимого операнда op1. Результат сохраняется в op1.

Флаги состояния

E	Z	V	C	N
*	*	0	0	*

- E Устанавливается, если содержимое op1 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Всегда сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
CPL R _w _n	91 n0	2

CPLB Поразрядная инверсия

Синтаксис CPLB op1

Действие $(op1) \leftarrow \neg(op1)$

Тип данных BYTE

Описание Вычисляется поразрядная инверсия содержимого операнда op1. Результат сохраняется в op1.

Флаги состояния

E	Z	V	C	N
*	*	0	0	*

- E Устанавливается, если содержимое op1 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Всегда сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
CPLB Rb _n	B1 n0	2

DISWDT Запрещение работы сторожевого таймера

Синтаксис DISWDT

Действие Запрещение работы сторожевого таймера

Описание Запрещается работа сторожевого таймера (WDT). WDT начинает работать после сброса. После сброса эту инструкцию следует выполнить до выполнения инструкции перезапуска сторожевого таймера (SRVWDT) или инструкции конца инициализации (EINIT). После выполнения одной из этих инструкций, DISWDT перестает действовать. Чтобы избежать случайного выполнения эта инструкция является защищенной.

Флаги состояния

<u>E</u>	Z	V	C	N
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
DISWDT	A5 5A A5 A5	4

DIV Деление со знаком (16 разрядов / 16 разрядов)

Синтаксис DIV op1

Действие (MDL) ← (MDL) / (op1)
(MDH) ← (MDL) mod (op1)

Тип данных WORD

Описание Выполняется деление содержимого 16-разрядного приемника MDL (младшее слово 32-разрядного регистра MD) на содержимое 16-разрядного источника op1. Оба операнда рассматриваются как числа в дополнительном коде. Частное сохраняется в MDL, остаток сохраняется в MDH (старшем слове регистра MD).
Инструкция DIV выполняется 20 машинных циклов. DIV является прерываемой инструкцией. Если во время выполнения DIV произошло прерывание, то устанавливается бит MULIP в регистре PSW и деление откладывается до выхода из обработчика прерывания.

Флаги состояния

E	Z	V	C	N
0	*	S	0	*

- E Всегда сбрасывается.
- Z Устанавливается, если результат равен нулю.
В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен в указанном типе данных или если содержимое делителя op1 было равно 0.
В противном случае сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
DIV Rw_n	4B nn	2

DIVL Деление со знаком (32 разряда / 16 разрядов)

Синтаксис DIVL op1

Действие (MDL) ← (MD) / (op1)
(MDH) ← (MD) mod (op1)

Тип данных WORD, DOUBLE WORD

Описание Выполняется деление содержимого 32-разрядного приемника MD на содержание 16-разрядного источника op1. Оба операнда рассматриваются как числа в дополнительном коде. Частное со знаком сохраняется в MDL (младшем слове регистра MD), остаток сохраняется в MDH (старшем слове регистра MD). Инструкция DIVL выполняется 20 машинных циклов. DIVL является прерываемой инструкцией. Если во время выполнения DIVL произошло прерывание, то устанавливается бит MULIP в регистре PSW и деление откладывается до выхода из обработчика прерывания.

Флаги состояния

E	Z	V	C	N
0	*	S	0	*

- E Всегда сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен в указанном типе данных или если содержимое делителя op1 было равно 0. В противном случае сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
DIVL Rw_n	6B nn	2

DIVLU Беззнаковое деление (32 разряда / 16 разрядов)

Синтаксис DIVLU op1

Действие (MDL) ← (MD) / (op1)
(MDH) ← (MD) mod (op1)

Тип данных WORD, DOUBLE WORD

Описание Выполняется беззнаковое деление содержимого 32-разрядного приемника MD на содержимое 16-разрядного источника op1. Частное сохраняется в MDL (младшем слове регистра MD), остаток сохраняется в MDH (старшем слове регистра MD). Инструкция DIVLU выполняется 20 машинных циклов. DIVLU является прерываемой инструкцией. Если во время выполнения DIVLU произошло прерывание, то устанавливается бит MULIP в регистре PSW и деление откладывается до выхода из обработчика прерывания.

Флаги состояния

E	Z	V	C	N
0	*	S	0	*

- E Всегда сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен в указанном типе данных или если содержимое делителя op1 было равно 0. В противном случае сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
DIVLU R _w _n	7B nn	2

DIVU Беззнаковое деление (16 разрядов / 16 разрядов)

Синтаксис DIVU op1

Действие (MDL) ← (MDL) / (op1)
(MDH) ← (MDL) mod (op1)

Тип данных WORD

Описание Выполняется беззнаковое деление содержимого 16-разрядного приемника (младшее слово регистра MD) на содержимое 16-разрядного источника op1. Частное сохраняется в MDL (младшем слове регистра MD), остаток сохраняется в MDH (старшем слове регистра MD).
Инструкция DIVU выполняется 20 машинных циклов. DIVU является прерываемой инструкцией. Если во время выполнения DIVU произошло прерывание, то устанавливается бит MULIP в регистре PSW и деление откладывается до выхода из обработчика прерывания.

Флаги состояния

E	Z	V	C	N
0	*	S	0	*

- E Всегда сбрасывается.
- Z Устанавливается, если результат равен нулю.
В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен в указанном типе данных или если содержимое делителя op1 было равно 0.
В противном случае сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
DIVU Rw_n	5B nn	2

EINIT Конец инициализации

Синтаксис EINIT

Действие Конец инициализации.

Описание После сброса на вывод микроконтроллера RSTOUT# выдается уровень лог.0, который сохраняется до выполнения EINIT, после чего выдается уровень лог.1. Это позволяет программе посылать внешним цепям микроконтроллера сигнал об успешной инициализации. После выполнения EINIT, инструкция запрещения сторожевого таймера DISWDT игнорируется. Инструкция EINIT используется для завершения инициализационной части программы. Чтобы избежать случайного выполнения эта инструкция является защищенной.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
EINIT	B5 4A B5 B5	4

EXTR Переадресация регистров

Синтаксис EXTR op1

Действие

```
(count) ← op1 [ 1 ≤ op1 ≤ 4 ]
IRQ_processing = Disable // запрещение обработки
                        // запросов на
                        // прерывание
SFR_range = Extended // перенаправление 8битовой
                    // адресации в область ESFR
DO WHILE ((count) ≠ 0 AND Class_B_IRQ ≠ TRUE)
    ... // выполнение следующей инструкции
    (count) ← (count) - 1
END WHILE
(count) = 0
SFR_range = Standard // отмена перенаправления
IRQ_processing = Enable // разрешение обработки
                  // запросов на
                  // прерывание
```

Описание При использовании адресации “reg”, “bitoff”, “bitaddr” производится обращение к расширенной области регистров специального назначения. На время выполнения указанного количества инструкций запрещается стандартная обработка запросов на прерывание от периферии и запросов класса A, а также обработка запросов от периферии контроллером PEC. Действие EXTR распространяется уже на следующую инструкцию, поэтому не требуется дополнительных инструкций NOP. Количество инструкций, на которые распространяется действие EXTR определяется op1 и принимает значение от 1 до 4 вне зависимости от количества требуемых циклов шины. Если во время выполнения указанного количества инструкций пришел запрос класса B, действие EXTR прекращается и осуществляется обработка запроса в соответствии со стандартной процедурой.

Замечание Устройства SAX-8XC166-X не поддерживают инструкцию EXTR.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника

EXTR #irang₂

Формат

D1 :10##-0

Размер

4

EXTP Страничная переадресация

Синтаксис EXTP op1, op2

Действие

```
(count) ← op2 [ 1 ≤ op1 ≤ 4 ]
IRQ_processing = Disable      // запрещение обработки
                               // запросов на
                               // прерывание
Data_Page = (op1)
DO WHILE ((count) ≠ 0 AND Class_B_IRQ ≠ TRUE)
    ... // выполнение следующей инструкции
    (count) ← (count) - 1
END WHILE
(count) = 0
Data_Page = (DPPx)
IRQ_processing = Enable      // разрешение обработки
                              // запросов на
                              // прерывание
```

Описание Заменяется стандартная схема адресации для режимов косвенной ([...]) и 16-битовой непосредственной адресаций (mem). При адресации 10-битовый номер страницы (разряды адреса A23...A14) определяется не содержимым соответствующего регистра DPPx, а значением op1. 14-битовое внутрестраничное смещение (разряды адреса A13...A0) определяется младшими разрядами адреса, указанного в инструкциях.

На время выполнения указанного количества инструкций запрещается стандартная обработка запросов на прерывание от периферии и запросов класса А, а также обработка запросов от периферии контроллером PEC. Действие EXTP распространяется уже на следующую инструкцию, поэтому не требуется дополнительных инструкций NOP. Количество инструкций, на которые распространяется действие EXTP определяется op2 и принимает значение от 1 до 4 вне зависимости от количества требуемых циклов шины. Если во время выполнения указанного количества инструкций пришел запрос класса В, действие EXTP прекращается и осуществляется обработка запроса в соответствии со стандартной процедурой.

Замечание Устройства SAX-8XC166-X не поддерживают инструкцию EXTP.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E Не изменяется.

Z Не изменяется.

V Не изменяется.

C Не изменяется.

N Не изменяется.

Формат инструкции

Мнемоника

Формат

Размер

EXTP $Rw_m, \#irang_2$

DC :01##-m 2

EXTP $\#pag_{10}, \#irang_2$

D7 :01##-0 PP 0:00PP 4

EXTPR Страничная переадресация и переадресация регистров

Синтаксис EXTPR op1, op2

Действие

```
(count) ← op2 [ 1 ≤ op2 ≤ 4 ]
IRQ_processing = Disable      // запрещение обработки
                               // запросов на
                               // прерывание
Data_Page = (op1)
SFR_range = Extended // перенаправление 8битовой
                    // адресации в область ESFR
DO WHILE ((count) ≠ 0 AND Class_B_IRQ ≠ TRUE)
    ... // выполнение следующей инструкции
    (count) ← (count) - 1
END WHILE
(count) = 0
Data_Page = (DPPx)
SFR_range = Standard // отмена перенаправления
IRQ_processing = Enable      // разрешение обработки
                               // запросов на
                               // прерывание
```

Описание

При использовании адресации “reg”, “bitoff”, “bitaddr” производится обращение к расширенной области регистров специального назначения. Кроме того, заменяется стандартная схема адресации для режимов косвенной ([...]) и 16-битовой непосредственной адресаций (mem). При адресации 10-битовый номер страницы (разряды адреса A23...A14) определяется не содержимым соответствующего регистра DPPx, а значением op1. 14-битовое внутривстраничное смещение (разряды адреса A13...A0) определяется младшими разрядами адреса, указанного в инструкциях. На время выполнения указанного количества инструкций запрещается стандартная обработка запросов на прерывание от периферии и запросов класса А, а также обработка запросов от периферии контроллером PEC. Действие EXTPR распространяется уже на следующую инструкцию, поэтому не требуется дополнительных инструкций NOP. Количество инструкций, на которые распространяется действие EXTPR определяется op2 и принимает значение от 1 до 4 вне зависимости от количества требуемых циклов шины. Если во время выполнения указанного количества инструкций пришел запрос класса В, действие EXTPR прекращается и осуществляется обработка запроса в соответствии со стандартной процедурой.

Замечание Устройства SAX-8XC166-X не поддерживают EXTPR.

Флаги состояния

E	Z	V	C	N
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника

Формат

Размер

EXTPR Rw_m , #irang₂

DC :11##-m

2

EXTPR #pag₁₀, #irang₂

D7 :11##-0 PP 0:00PP

4

EXTS Сегментная переадресация

Синтаксис EXTS op1

Действие

```
(count) ← op1 [ 1 ≤ op1 ≤ 4 ]
IRQ_processing = Disable      // запрещение обработки
                               // запросов на
                               // прерывание
Data_Segment = (op1)
DO WHILE ((count) ≠ 0 AND Class_B_IRQ ≠ TRUE)
    ... // выполнение следующей инструкции
    (count) ← (count) - 1
END WHILE
(count) = 0
Data_Page = (DPPx)
IRQ_processing = Enable      // разрешение обработки
                               // запросов на
                               // прерывание
```

Описание

Заменяет стандартную схему адресации для режимов косвенной ([...]) и 16-битовой непосредственной адресаций (mem). При адресации 8-битовый номер сегмента (разряды адреса A23...A16) определяется значением op1. 16-битовое внутрисегментное смещение (разряды адреса A15...A0) определяется адресом, указанным в инструкциях. На время выполнения указанного количества инструкций запрещается стандартная обработка запросов на прерывание от периферии и запросов класса А, а также обработка запросов от периферии контроллером PEC. Действие EXTS распространяется уже на следующую инструкцию, поэтому не требуется дополнительных инструкций NOP. Количество инструкций, на которые распространяется действие EXTS определяется op2 и принимает значение от 1 до 4 вне зависимости от количества требуемых циклов шины. Если во время выполнения указанного количества инструкций пришел запрос класса В, действие EXTS прекращается и осуществляется обработка запроса в соответствии со стандартной процедурой.

Замечание Устройства SAX-8XC166-X не поддерживают инструкцию EXTS.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E Не изменяется.
Z Не изменяется.
V Не изменяется.
C Не изменяется.
N Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
EXTS $Rw_m, #irang$	DC :00##-m	2
EXTS $\#seg_8, #irang_2$	D7 :00##-0 SS 00	4

EXTSR Сегментная переадресация и переадресация регистров

Синтаксис EXTSR op1, opt2

Действие

```
(count) ← op2 [ 1 ≤ op2 ≤ 4 ]
IRQ_processing = Disable      // запрещение обработки
                               // запросов на
                               // прерывание
Data_Segment = (op1)
SFR_range = Extended // перенаправление 8битовой
                  // адресации в область ESFR
DO WHILE ((count) ≠ 0 AND Class_B_IRQ ≠ TRUE))
    ... // выполнение следующей инструкции
    (count) (count) - 1
END WHILE
(count) = 0
Data_Page = (DPPx)
SFR_range = Standard // отмена перенаправления
IRQ_processing = Enable      // разрешение обработки
                              // запросов на
                              // прерывание
```

Описание При использовании адресации “reg”, “bitoff”, “bitaddr” производится обращение к расширенной области регистров специального назначения. Кроме того, заменяется стандартная схема адресации для режимов косвенной ([...]) и 16-битовой непосредственной адресаций (mem). При адресации 8-битовый номер сегмента (разряды адреса A23...A16) определяется значением op1. 16-битовое внутрисегментное смещение (разряды адреса A15...A0) определяется адресом, указанным в инструкциях.

На время выполнения указанного количества инструкций запрещается стандартная обработка запросов на прерывание от периферии и запросов класса А, а также обработка запросов от периферии контроллером PEC. Действие EXTISR распространяется уже на следующую инструкцию, поэтому не требуется дополнительных инструкций NOP. Количество инструкций, на которые распространяется действие EXTISR определяется op2 и принимает значение от 1 до 4 вне зависимости от количества требуемых циклов шины. Если во время выполнения указанного количества инструкций пришел запрос класса В, действие EXTISR прекращается и осуществляется обработка запроса в соответствии со стандартной процедурой.

Замечание Устройства SAX-8XC166-X не поддерживают инструкцию EXTISR.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E Не изменяется.

Z Не изменяется.

V Не изменяется.

C Не изменяется.

N Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
EXTSR $Rw_m, \#irang_2$	DC :10##-m	2
EXTSR $\#seg_8, \#irang_2$	D7 :10##-0 SS 00	4

IDLE Режим ожидания

Синтаксис IDLE

Действие Переход в режим ожидания.

Описание Осуществляет переход в режим ожидания. В этом режиме ЦПУ останавливается, в то время как периферийные устройства продолжают работу. Выход из режима осуществляется по прерыванию от внутрикристальных периферийных устройств или по внешнему прерыванию. Чтобы избежать случайного выполнения эта инструкция реализована в виде защищенной.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника

IDLE

Формат

87 78 87 87

Размер

4

JB Относительный переход, если бит установлен

Синтаксис JB op1, op2

Действие IF (op1) = 1 THEN
 (IP) ← (IP) + op2 * 2
 ELSE
 ... // выполнение следующей инструкции
 END IF

Тип данных BIT

Описание Если содержимое op1 равно 1, то осуществляется переход внутри сегмента по адресу, определяемому суммой содержимого указателя инструкций IP и удвоенного смещения op2. Смещение воспринимается как число в дополнительном коде. Для вычисления адреса перехода используется значение IP, равное адресу инструкции, следующей за JB. Если указанный бит равен 0, то выполняется инструкция, следующая за JB.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
JB bitaddr _{q,q} , rel	8A QQ rr q0	4

JBC Относительный переход с очисткой бита, если бит установлен

Синтаксис JBC op1, op2

Действие

```
IF (op1) = 1 THEN
    (op1) = 0
    (IP) ← (IP) + op2 * 2
ELSE
    ... // выполнение следующей инструкции
END IF
```

Тип данных BIT

Описание Если содержимое op1 равно 1, то op1 обнуляется и осуществляется переход внутри сегмента по адресу, определяемому суммой содержимого указателя инструкций IP и удвоенного смещения op2. Смещение воспринимается как число в дополнительном коде. Для вычисления адреса перехода используется значение IP, равное адресу инструкции, следующей за JBC. Если указанный бит равен 0, выполняется инструкция, следующая за JBC.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
0	B#	0	0	B

- E Всегда сбрасывается.
- Z Содержит инверсное значение предыдущего состояния указанного бита.
- V Всегда сбрасывается.
- C Всегда сбрасывается.
- N Содержит предыдущее состояние указанного бита.

Формат инструкции

Мнемоника	Формат	Размер
JBC bitaddr _q , rel	AA QQ rr q0	4

JMPA Абсолютный условный переход

Синтаксис JMPA op1, op2

Действие IF (op1) = 1 THEN
 (IP) ← op2
 ELSE
 ... // выполнение следующей инструкции
 END IF

Описание Если выполняется условие, указанное в op1, осуществляется переход внутри сегмента по адресу, указанному в op2. Если условие не выполняется, никаких действий не производится и выполняется инструкция, следующая за JMPA.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
JMPA cc, caddr	EA c0 MM MM	4

JMPI Косвенный условный переход

Синтаксис JMPI op1, op2

Действие IF (op1) = 1 THEN
 (IP) ← (op2)
 ELSE
 ... // выполнение следующей инструкции
 END IF

Описание Если выполняется условие, указанное в op1, осуществляется переход внутри сегмента по адресу, равному содержимому op2. Если условие не выполняется, никаких действий не производится и выполняется инструкция, следующая за JMPI.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
JMPI cc, [Rw _n]	9C cn	2

JMPR Относительный условный переход

Синтаксис JMPR op1, op2

Действие IF (op1) = 1 THEN
 (IP) ← (IP) + op2 * 2
 ELSE
 ... // выполнение следующей инструкции
 END IF

Описание Если выполняется условие, указанное в op1, то осуществляется переход внутри сегмента по адресу, определяемому суммой указателя инструкций IP и удвоенного смещения, указанного в op2. Смещение воспринимается как число в дополнительном коде. Для вычисления адреса перехода используется значение IP, равное адресу инструкции, следующей за JMPR. Если условие не выполняется, никаких действий не производится и выполняется инструкция, следующая за JMPR.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
JMPR cc, rel	cD rr	2

JMPS Абсолютный межсегментный переход

Синтаксис JMPS op1, op2

Действие (CSP) ← op1
(IP) ← op2

Описание Осуществляется безусловный переход по полному 24-битовому адресу. Номер сегмента определяется операндом op1, а внутрисегментное смещение – операндом op2.

Флаги состояния

E	Z	V	C	N
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
JMPS seg, caddr	FA SS MM MM	4

JNB Относительный переход, если бит сброшен

Синтаксис JNB op1, op2

Действие IF (op1) = 1 THEN
 (IP) ← (IP) + op2 * 2
 ELSE
 ... // выполнение следующей инструкции
 END IF

Тип данных BIT

Описание Если содержимое op1 равно 0, то осуществляется переход внутри сегмента по адресу, определяемому суммой содержимого указателя инструкций IP и удвоенного смещения op2. Смещение воспринимается как число в дополнительном коде. Для вычисления адреса перехода используется значение IP, равное адресу инструкции, следующей за JNB. Если указанный бит равен 1, выполняется инструкция, следующая за JNB.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
JNB bitaddr _q , rel	9A QQ rr q0	4

JNBS Относительный переход с установкой бита, если бит сброшен

Синтаксис JNBS op1, op2

Действие

```
IF (op1) = 0 THEN
    (op1) = 1
    (IP) ← (IP) + op2 * 2
ELSE
    ... // выполнение следующей инструкции
END IF
```

Тип данных BIT

Описание Если содержимое op1 равно 0, то осуществляется переход внутри сегмента по адресу, определяемому суммой содержимого указателя инструкций IP и удвоенного смещения op2. Смещение воспринимается как число в дополнительном коде. Бит, указанный в op1, перед переходом устанавливается в 1. Для вычисления адреса перехода используется значение IP, равное адресу инструкции, следующей за JNBS. Если указанный бит равен 1, выполняется инструкция, следующая за JNBS.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	B#	-	-	B

- E Всегда сбрасывается.
- Z Содержит инверсное значение предыдущего состояния указанного бита.
- V Всегда сбрасывается.
- C Всегда сбрасывается.
- N Содержит предыдущее состояние указанного бита.

Формат инструкции

Мнемоника	Формат	Размер
JNBS bitaddr _{0..q} , rel	BA QQ rr q0	4

MOV Пересылка данных

Синтаксис MOV op1, op2

Действие (op1) ← (op2)

Тип данных WORD

Описание Содержимое источника op2 пересылается в приемник op1. Пересылаемые данные анализируются и выставляются флаги состояния.

Флаги состояния

E	Z	V	C	N
*	*	-	-	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если содержимое op2 равно нулю. В противном случае сбрасывается.
- V Не изменяется.
- C Не изменяется.
- N Устанавливается, если установлен старший значащий разряд содержимого op2. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
MOV Rw_n, Rw_m	F0 nm	2
MOV $Rw_n, \#data_4$	E0 #n	2
MOV $reg, \#data_{16}$	E6 RR ## ##	4
MOV $Rw_n, [Rw_m]$	A8 nm	2
MOV $Rw_n, [Rw_m+]$	98 nm	2
MOV $[Rw_m], Rw_n$	B8 nm	2
MOV $[-Rw_m], Rw_n$	88 nm	2
MOV $[Rw_n], [Rw_m]$	C8 nm	2
MOV $[Rw_n+], [Rw_m]$	D8 nm	2
MOV $[Rw_n], [Rw_m+]$	E8 nm	2
MOV $Rw_n, [Rw_m+\#data_{16}]$	D4 nm ## ##	4
MOV $[Rw_m+\#data_{16}], Rw_n$	C4 nm ## ##	4
MOV $[Rw_n], mem$	84 0n MM MM	4
MOV $mem, [Rw_n]$	94 0n MM MM	4
MOV reg, mem	F2 RR MM MM	4
MOV mem, reg	F6 RR MM MM	4

MOVB Пересылка данных

Синтаксис MOVB op1, op2

Действие (op1) ← (op2)

Тип данных BYTE

Описание Пересылается содержимое источника op2 в приемник op1. Пересылаемые данные анализируются и выставляются флаги состояния.

Флаги состояния

E	Z	V	C	N
*	*	-	-	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если содержимое op2 равно нулю. В противном случае сбрасывается.
- V Не изменяется.
- C Не изменяется.
- N Устанавливается, если установлен старший значащий разряд содержимого op2. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
MOVB Rb _n , Rb _m	F1 nm	2
MOVB Rb _n , #data ₄	E1 #n	2
MOVB reg, #data ₈	E7 RR ## xx	4
MOVB Rb _n , [Rw _m]	A9 nm	2
MOVB Rb _n , [Rw _m +]	99 nm	2
MOVB [Rw _m], Rb _n	B9 nm	2
MOVB [-Rw _m], Rb _n	89 nm	2
MOVB [Rw _n], [Rw _m]	C9 nm	2
MOVB [Rw _n +] , [Rw _m]	D9 nm	2
MOVB [Rw _n], [Rw _m +]	E9 nm	2
MOVB Rb _n , [Rw _m +#data ₁₆]	F4 nm ## ##	4
MOVB [Rw _m +#data ₁₆], Rb _n	E4 nm ## ##	4
MOVB [Rw _n], mem	A4 0n MM MM	4
MOVB mem, [Rw _n]	B4 0n MM MM	4
MOVB reg, mem	F3 RR MM MM	4
MOVB mem, reg	F7 RR MM MM	4

MOVBS Пересылка байта с расширением знака

Синтаксис MOVBS op1, op2

Действие (low byte op1) ← (op2)
 IF (op2₇) = 1 THEN
 (high byte (op1)) ← FFh
 ELSE
 (high byte (op1)) ← 00h
 END IF

Тип данных WORD, BYTE

Описание Пересылается байт содержимого источника op2 в младший байт приемника op1. Пересылаемые данные анализируются и выставляются флаги состояния. Старший байт приемника op1 заполняется знаковым (старшим) битом источника op2.

Флаги состояния

E	Z	V	C	N
0	*	-	-	0

- E Всегда сбрасывается.
- Z Устанавливается, если содержимое op2 равно нулю. В противном случае сбрасывается.
- V Не изменяется.
- C Не изменяется.
- N Устанавливается, если установлен старший значащий разряд содержимого op2. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
MOVBS R _{w_n} , R _{b_m}	D0 mn	2
MOVBS reg, mem	D2 RR MM MM	4
MOVBS mem, reg	D5 RR MM MM	4

MOVBSZ Пересылка байта с очисткой старшего байта

Синтаксис MOVBSZ op1, op2

Действие (low byte (op1)) ← (op2)
(high byte (op1)) ← 00h

Тип данных WORD, BYTE

Описание Пересылается байт содержимого источника op2 в младший байт приемника op1. Пересылаемые данные анализируются и выставляются флаги состояния. Старший байт приемника op1 заполняется нулями.

Флаги состояния

E	Z	V	C	N
0	*	-	-	0

E Всегда сбрасывается.

Z Устанавливается, если содержимое источника op2 равно нулю. В противном случае сбрасывается.

V Не изменяется.

C Не изменяется.

N Всегда сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
MOVBSZ R _{w_n} , R _{b_m}	C0 mn	2
MOVBSZ reg, mem	C2 RR MM MM	4
MOVBSZ mem, reg	C5 RR MM MM	4

MUL Умножение со знаком

Синтаксис MUL op1, op2

Действие (MD) \leftarrow (op1) * (op2)

Тип данных WORD

Описание Выполняется умножение содержимого двух 16-разрядных операндов op1 и op2. Оба операнда рассматриваются как числа в дополнительном коде. Результат (32 разряда) сохраняется в регистре MD.
Инструкция MUL выполняется 10 машинных циклов. MUL является прерываемой инструкцией. Если во время выполнения MUL произошло прерывание, то устанавливается бит MULIP в регистре PSW и умножение откладывается до выхода из обработчика прерывания.

Флаги состояния

E	Z	V	C	N
0	*	S	0	0

- E Всегда сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Устанавливается, если результат не может быть представлен 16-разрядным числом. В противном случае сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
MUL Rw_n, Rw_m	0B nm	2

MULU Беззнаковое умножение

Синтаксис MULU op1, op2

Действие (MD) ← (op1) * (op2)

Тип данных WORD

Описание Выполняется беззнаковое умножение содержимого двух 16-разрядных операндов op1 и op2. Результат (32 разряда) сохраняется в регистре MD. Инструкция MULU выполняется 10 машинных циклов. MULU является прерываемой инструкцией. Если во время выполнения MULU произошло прерывание, то устанавливается бит MULIP в регистре PSW и умножение откладывается до выхода из обработчика прерывания.

Флаги состояния

E	Z	V	C	N
0	*	S	0	*

- E Всегда сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Устанавливается, если результат не может быть представлен 16-разрядным числом. В противном случае сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
MULU Rw_n, Rw_m	1B nm	2

NEG Инверсия знака

Синтаксис NEG op1

Действие (op1) ← 0 - (op1)

Тип данных WORD

Описание Выполняется изменение знака операнда op1. Результат сохраняется в op1.

Флаги состояния

E	Z	V	C	N
*	*	*	S	*

- E Устанавливается, если содержимое op1 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен в указанном типе данных. В противном случае сбрасывается.
- C Устанавливается, если произошел заем. В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий бит результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
NEG Rw_n	81 n0	2

NEGB Инверсия знака

Синтаксис NEGB op1

Действие (op1) ← 0 - (op1)

Тип данных BYTE

Описание Выполняется изменение знака операнда op1. Результат сохраняется в op1.

Флаги состояния

E	Z	V	C	N
*	*	*	S	*

- E Устанавливается, если содержимое op1 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен в указанном типе данных. В противном случае сбрасывается.
- C Устанавливается, если произошел заем. В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
NEGB Rb _n	A1 n0	2

NOP Нет операции

Синтаксис NOP

Действие Пустая команда.

Описание Эта инструкция не вызывает никаких действий и не влияет на флаги.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E Не изменяется.

Z Не изменяется.

V Не изменяется.

C Не изменяется.

N Не изменяется.

Формат инструкции

Мнемоника

NOP

Формат

CC 00

Размер

2

OR Логическое ИЛИ

Синтаксис OR op1, op2

Действие $(op1) \leftarrow (op1) \vee (op2)$

Тип данных WORD

Описание Выполняется побитовая операция логического ИЛИ над содержимым операндов источника op2 и приемника op1. Результат сохраняется в op1.

Флаги состояния

E	Z	V	C	N
*	*	0	0	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю.
В противном случае сбрасывается.
- V Всегда сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
OR Rw_n, Rw_m	70 nm	2
OR $Rw_n, [Rw_1]$	78 n:10ii	2
OR $Rw_n, [Rw_1+]$	78 n:11ii	2
OR $Rw_n, \#data_3$	78 n:0###	2
OR $reg, \#data_{16}$	76 RR ## ##	4
OR reg, mem	72 RR MM MM	4
OR mem, reg	74 RR MM MM	4

ORB Логическое ИЛИ

Синтаксис ORB op1, op2

Действие $(op1) \leftarrow (op1) \vee (op2)$

Тип данных BYTE

Описание Выполняется побитовая операция логического ИЛИ над содержимым операндов источника op2 и приемника op1. Результат сохраняется в op1.

Флаги состояния

E	Z	V	C	N
*	*	0	0	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю.
В противном случае сбрасывается.
- V Всегда сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
ORB Rb _n , Rb _m	71 nm	2
ORB Rb _n , [Rw ₁]	79 n:10ii	2
ORB Rb _n , [Rw ₁ +]	79 n:11ii	2
ORB Rb _n , #data ₃	79 n:0###	2
ORB reg, #data ₈	77 RR ## xx	4
ORB reg, mem	73 RR MM MM	4
ORB mem, reg	75 RR MM MM	4

PCALL Абсолютный вызов подпрограмм с сохранением слова на стеке

Синтаксис PCALL op1, op2

Действие

```
(tmp) ← (op1)
(SP) ← (SP) - 2
((SP)) ← (tmp)
(SP) ← (SP) - 2
((SP)) ← (IP)
(IP) ← op2
```

Тип данных WORD

Описание На вершину системного стека помещается содержимое операнда op1 и содержимое указателя инструкций IP и осуществляется переход по адресу, определяемому операндом op 2. Значение указателя стека SP перед каждым занесением на стек уменьшается на 2. Поскольку IP всегда указывает на инструкцию, следующую за PCALL, значение, сохраненное в стеке, представляет собой адрес возврата для вызываемой подпрограммы.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
*	*	-	-	*

- E Устанавливается, если содержимое операнда op1 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если содержимое операнда op1 равно нулю. В противном случае сбрасывается.
- V Не изменяется.
- C Не изменяется.
- N Устанавливается, если установлен старший значащий разряд содержимого op1. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
PCALL reg, caddr	E2 RR MM MM	4

POP Извлечь слово из системного стека

Синтаксис POP op1

Действие

$$(tmp) \leftarrow ((SP))$$

$$(SP) \leftarrow (SP) + 2$$

$$(op1) \leftarrow (tmp)$$

Тип данных WORD

Описание С вершины системного стека извлекается значение и помещается в приемник в op1. Затем указатель стека SP увеличивается на 2.

Флаги состояния

E	Z	V	C	N
*	*	-	-	*

- E Устанавливается, если извлекаемое значение равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если значение извлекаемого слова равно нулю. В противном случае сбрасывается.
- V Не изменяется.
- C Не изменяется.
- N Устанавливается, если установлен старший значащий разряд извлекаемого значения. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
POP reg	FC RR	2

PRIOR Счетчик нормализации

Синтаксис PRIOR op1, op2

Действие $(tmp) \leftarrow (op2)$
 $(count) \leftarrow 0$
 DO WHILE $((tmp_{15}) \neq 1 \text{ AND } (count) \neq 15 \text{ AND } (op2) \neq 0)$
 $(tmp_n) \leftarrow (tmp_{n-1})$
 $(count) \leftarrow (count) + 1$
 END WHILE
 $(op1) \leftarrow (count)$

Тип данных WORD

Описание Вычисляется значение счетчика op1, показывающее количество битовых сдвигов в сторону старших разрядов, требуемых для нормализации содержимого операнда op2, (чтобы его старший значащий бит после сдвига был равен 1). Если содержимое op2 равно нулю, то содержимое op1 обнуляется и устанавливается флаг Z, в противном случае флаг Z сбрасывается. Эта инструкция может использоваться для реализации вычислений с плавающей точкой.

Флаги состояния

E	Z	V	C	N
0	*	0	0	0

- E Всегда сбрасывается.
- Z Устанавливается, если содержимое op2 равно нулю. В противном случае сбрасывается.
- V Всегда сбрасывается.
- C Всегда сбрасывается.
- N Всегда сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
PRIOR Rw_n, Rw_m	2B nm	2

PUSH Поместить слово на стек

Синтаксис PUSH op1

Действие (tmp) ← (op1)
 (SP) ← (SP) - 2
 ((SP)) ← (tmp)

Тип данных WORD

Описание Помещает содержимое op1 на вершину системного стека после уменьшения указателя стека SP на 2.

Флаги состояния

E	Z	V	C	N
*	*	-	-	*

- E Устанавливается, если помещаемое значение равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если помещаемое слово равно нулю. В противном случае сбрасывается.
- V Не изменяется.
- C Не изменяется.
- N Устанавливается, если установлен старший значащий разряд помещаемого значения. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
PUSH reg	EC RR	2

PWRDN Режим пониженного потребления

Синтаксис PWRDN

Действие Микроконтроллер входит в режим пониженного потребления.

Описание Осуществляется переход всей внутренней периферии и ЦПУ в режим пониженного энергопотребления до прихода сигнала внешнего сброса микроконтроллера. Для защиты от неправильного выполнения эта инструкция является защищенной.
Выполнение команды PWRDN разрешено, когда к контакту немаскируемого запроса на прерывание NMI# приложено напряжение уровня лог.0, в противном случае команда не выполняется.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
PWRDN	97 68 97 97	4

RET Возврат из подпрограммы

Синтаксис RET

Действие (IP) ← ((SP))
(SP) ← (SP) + 2

Описание Осуществляется возврат из подпрограммы. С вершины системного стека извлекается значение и помещается в указатель инструкций IP для адресации следующей исполняемой инструкции. После извлечения указатель стека увеличивается на 2. Возврат из подпрограммы, вызванной с помощью CALLI, CALLR или CALLA, осуществляется на инструкцию, следующую за вызовом.

Флаги состояния

E	Z	V	C	N
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
RET	CB 00	2

RETI Возврат из подпрограммы обработки прерывания

Синтаксис RETI

Действие

```
(IP) ← ((SP))
(SP) ← (SP) + 2
IF (SYSCON.SGDTDIS = 0) THEN
    (CSP) ← ((SP))
    (SP) ← (SP) + 2
END IF
(PSW) ← ((SP))
(SP) ← (SP) + 2
```

Описание Осуществляется возврат из подпрограммы обработки прерывания. С вершины системного стека извлекаются слова и помещаются в IP, CSP и PSW. После каждого извлечения указатель стека SP увеличивается на 2. Выполнение продолжается с инструкции, следующей за той, во время выполнения которой пришел запрос на прерывание (см. главу “Прерывания” документации на микроконтроллер). Предыдущее состояние ЦПУ восстанавливается после извлечения PSW. Значение CSP извлекается только, если разрешена сегментация (бит SGDTDIS в регистре SYSCON равен нулю).

Флаги состояния

E	Z	V	C	N
S	S	S	S	S

E	Восстанавливается при извлечении PSW из стека.
Z	Восстанавливается при извлечении PSW из стека.
V	Восстанавливается при извлечении PSW из стека.
C	Восстанавливается при извлечении PSW из стека.
N	Восстанавливается при извлечении PSW из стека.

Формат инструкции

Мнемоника	Формат	Размер
RETI	FB 88	2

RETP Возврат из подпрограммы и извлечение слова

Синтаксис RETP op1

Действие

$$(IP) \leftarrow ((SP))$$

$$(SP) \leftarrow (SP) + 2$$

$$(tmp) \leftarrow ((SP))$$

$$(SP) \leftarrow (SP) + 2$$

$$(op1) \leftarrow (tmp)$$

Тип данных WORD

Описание Осуществляется возврат из подпрограммы. С вершины системного стека извлекается слово и помещается в указатель инструкций IP для адресации следующей исполняемой инструкции. После извлечения указатель стека увеличивается на 2. Затем с измененной вершины системного стека извлекается значение и записывается в приемник op1, указатель стека еще раз увеличивается на 2. Возврат из подпрограммы, вызванной с помощью PCALL, осуществляется на инструкцию, следующую за вызовом.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
*	*	-	-	*

- E Устанавливается, если извлекаемое в op1 значение равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если значение извлекаемого слова равно нулю. В противном случае сбрасывается.
- V Не изменяется.
- C Не изменяется.
- N Устанавливается, если установлен старший значащий разряд извлекаемого в op1 значения. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
RETP reg	EB RR	2

RETS Межсегментный возврат из подпрограммы

Синтаксис RETS

Действие $(IP) \leftarrow ((SP))$
 $(SP) \leftarrow (SP) + 2$
 $(CSP) \leftarrow ((SP))$
 $(SP) \leftarrow (SP) + 2$

Описание Осуществляется межсегментный возврат из подпрограммы. С вершины системного стека извлекаются два значения и помещаются в указатель инструкций IP и указатель номера сегмента CSP для адресации следующей исполняемой инструкции. После каждого извлечения указатель стека увеличивается на 2. Возврат из подпрограммы, вызванной с помощью CALLS, осуществляется на инструкцию, следующую за вызовом.

Флаги состояния

E	Z	V	C	N
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
RETS	DB 00	2

ROL Циклический сдвиг в сторону старших разрядов

Синтаксис ROL op1, op2

Действие

```
(count) ← (op2)
(C) ← 0
DO WHILE ((count) ≠ 0)
    (C) ← (op115)
    (op1n) ← (op1n-1) [n = 1...15]
    (op10) ← (C)
    (count) ← (count) - 1
END WHILE
```

Тип данных WORD

Описание Осуществляется циклический сдвиг содержимого op1 в сторону старших разрядов на количество позиций, определяемое операндом op2. Разряд 15 циклически сдвигается в разряд 0 и во флаг переноса C. Допустимые значения количества сдвигов – от 0 до 15 включительно. Если количество сдвигов определяется содержимым PОН, то используется только четыре младших разряда.

Флаги состояния

E	Z	V	C	N
0	*	0	S	*

- E Всегда сбрасывается.
- Z Устанавливается если результат равен нулю. В противном случае сбрасывается.
- V Всегда сбрасывается.
- C Устанавливается в значение последнего выдвинутого из op1 старшего значащего разряда. Сбрасывается при нулевом количестве позиций.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
ROL R _{w_n} , R _{w_m}	0C nm	2
ROL R _{w_n} , #data ₄	1C #n	2

ROR Циклический сдвиг в сторону младших разрядов

Синтаксис ROR op1, op2

Действие

```
(count) ← (op2)
(C) ← 0
(V) ← 0
DO WHILE ((count) ≠ 0)
  (V) ← (V) ∨ (C)
  (C) ← (op10)
  (op1n) ← (op1n+1) [n = 0...14]
  (op115) ← (C)
  (count) ← (count) - 1
END WHILE
```

Тип данных WORD

Описание Осуществляется циклический сдвиг содержимого op1 в сторону младших разрядов на количество позиций, определяемое операндом op2. Разряд 0 циклически сдвигается в разряд 15 и во флаг переноса C. Допустимые значения количества сдвигов – от 0 до 15 включительно. Если количество сдвигов определяется содержимым PОН, то используется только четыре младших разряда.

Флаги состояния

E	Z	V	C	N
0	*	S	S	*

- E Всегда сбрасывается.
- Z Устанавливается, если результат равен нулю. В противном случае сбрасывается.
- V Устанавливается, если на любом этапе сдвига значение 1 выдвигается из флага переноса C. Сбрасывается при нулевом количестве позиций.
- C Устанавливается в значение последнего выдвинутого из op1 старшего значащего бита. Сбрасывается при нулевом количестве позиций.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
ROR R _{w_n} , R _{w_m}	2C nm	2
ROR R _{w_n} , #data ₄	3C #n	2

SCXT Переключение контекста

Синтаксис SCXT op1, op2

Действие

```
(tmp1) ← (op1)
(tmp2) ← (op2)
(SP) ← (SP) - 2
((SP)) ← (tmp1)
(op1) ← (tmp2)
```

Описание Осуществляется сохранение на вершине стека содержимого регистра и загрузка регистра новым значением. После сохранения, перед загрузкой, указатель стека уменьшается на 2. Эта инструкция используется в основном для смены банка РОН и сохранения MDC обработчиком прерывания.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
SCXT reg, #data ₁₆	C6 RR ## ##	4
SCXT reg, mem	D6 RR MM MM	4

SHL Сдвиг в сторону старших адресов

Синтаксис SHL op1, op2

Действие

```
(count) ← (op2)
(C) ← 0
DO WHILE ((count) ≠ 0)
  (C) ← (op115)
  (op1n) ← (op1n-1) [n = 1...15]
  (op10) ← 0
  (count) ← (count) - 1
END WHILE
```

Тип данных WORD

Описание Осуществляется сдвиг содержимого op1 в сторону старших разрядов на количество позиций, определяемое операндом op2. Младшие разряды op1 заполняются нулями. Старший значащий разряд выдвигается во флаг переноса C. Допустимые значения количества сдвигов – от 0 до 15 включительно. Если количество сдвигов определяется содержимым POH, то используется только четыре младших разряда.

Флаги состояния

E	Z	V	C	N
0	*	0	S	*

- E Всегда сбрасывается.
- Z Устанавливается если результат равен нулю. В противном случае сбрасывается.
- V Всегда сбрасывается.
- C Устанавливается в значение последнего выдвинутого из op1 старшего значащего разряда. Сбрасывается при нулевом количестве позиций.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

SHL	Rw _n , Rw _m	4C nm	2
SHL	Rw _n , #data ₄	5C #n	2

SHR Сдвиг в сторону младших адресов

Синтаксис SHR op1, op2

Действие

```
(count) ← (op2)
(C) ← 0, (V) ← 0
DO WHILE ((count) ≠ 0)
    (V) ← (C) ∨ (V)
    (C) ← (op10), (op1n) ← (op1n+1) [n = 0...14]
    (op115) ← 0
    (count) ← (count) - 1
END WHILE
```

Тип данных WORD

Описание Осуществляется сдвиг содержимого op1 в сторону младших разрядов на количество позиций, определяемое операндом op2. Старшие разряды op1 заполняются нулями. Поскольку выдвигаемые биты представляют остаток, флаг переполнения V используется как флаг округления. Этот флаг совместно с флагом переноса C помогает пользователю определить, были ли потерянные биты остатка больше, меньше или равны половине младшего значащего бита. Допустимые значения количества сдвигов – от 0 до 15 включительно. Если количество сдвигов определяется содержимым PОН, то используется только четыре младших разряда.

Флаги состояния

E	Z	V	C	N
0	*	S	S	*

- E Всегда сбрасывается.
- Z Устанавливается если результат равен 0. В противном случае сбрасывается.
- V Устанавливается если в любом цикле операции сдвига из флага переноса выдвигается 1. Сбрасывается при нулевом количестве сдвигов.
- C Устанавливается в значение последнего выдвинутого из op1 старшего значащего бита. Сбрасывается при нулевом количестве сдвигов.
- N Устанавливается, если установлен старший значащий бит результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
SHR Rw_n, Rw_m	6C nm	2
SHR $Rw_n, \#data_4$	7C #n	2

SRST Программный сброс

Синтаксис SRST

Действие Программный сброс.

Описание Выполняется программный сброс микроконтроллера. Программный сброс аналогичен внешнему аппаратному сбросу. Во избежание случайного выполнения эта инструкция является защищенной.

Флаги состояния

E	Z	V	C	N
0	0	0	0	0

- E Всегда сбрасывается.
- Z Всегда сбрасывается.
- V Всегда сбрасывается.
- C Всегда сбрасывается.
- N Всегда сбрасывается.

Формат инструкции

Мнемоника

SRST

Формат

B7 48 B7 B7

Размер

4

SRVWDT Перезапуск сторожевого таймера

Синтаксис SRVWDT

Действие Перезапуск сторожевого таймера.

Описание Перезагружается старший байт счетчика сторожевого таймера значением из регистра WDTCON и очищается младший байт. После выполнения данной инструкции невозможно запретить работу сторожевого таймера до следующего сброса микроконтроллера. Во избежание случайного выполнения эта инструкция является защищенной.

Флаги состояния

<u>E</u>	<u>Z</u>	<u>V</u>	<u>C</u>	<u>N</u>
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника

SRVWDT

Формат

A7 58 A7 A7

Размер

4

SUB Целочисленное вычитание

Синтаксис SUB op1, op2

Действие (op1) ← (op1) - (op2)

Тип данных WORD

Описание Выполняется двоичное вычитание в дополнительном коде содержимого источника op2 из содержимого приемника op1. Результат сохраняется в op1.

Флаги состояния

E	Z	V	C	N
*	*	*	S	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу.
В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю.
В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен указанным типом данных. В противном случае сбрасывается.
- C Устанавливается, если произошел заем.
В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
SUB Rw_n, Rw_n	20 nm	2
SUB $Rw_n, [Rw_i]$	28 n:10ii	2
SUB $Rw_n, [Rw_i+]$	28 n:11ii	2
SUB $Rw_n, \#data_3$	28 n:0###	2
SUB $reg, \#data_{16}$	26 RR ## ##	4
SUB reg, mem	22 RR MM MM	4
SUB mem, reg	24 RR MM MM	4

SUBB Целочисленное вычитание

Синтаксис SUBB op1, op2

Действие $(op1) \leftarrow (op1) - (op2)$

Тип данных BYTE

Описание Выполняется двоичное вычитание в дополнительном коде содержимого источника op2 из содержимого приемника op1. Результат сохраняется в op1.

Флаги состояния

E	Z	V	C	N
*	*	*	S	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу.
В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю.
В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен указанным типом данных. В противном случае сбрасывается.
- C Устанавливается, если произошел заем.
В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
SUBB Rb _n , Rb _m	21 nm	2
SUBB Rb _n , [Rw _i]	29 n:10ii	2
SUBB Rb _n , [Rw _i +]	29 n:11ii	2
SUBB Rb _n , #data ₃	29 n:0###	2
SUBB reg, #data ₈	27 RR ## xx	4
SUBB reg, mem	23 RR MM MM	4
SUBB mem, reg	25 RR MM MM	4

SUBC Целочисленное вычитание с переносом

Синтаксис SUBC op1, op2

Действие (op1) ← (op1) - (op2) - (C)

Тип данных WORD

Описание Выполняется вычитание содержимого источника op2 и бита переноса C из содержимого приемника op1. Результат сохраняется в op1. Эта инструкция используется для реализации вычислений с повышенной точностью.

Флаги состояния

E	Z	V	C	N
*	S	*	S	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю и перед выполнением команды был установлен флаг Z. В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен указанным типом данных. В противном случае сбрасывается.
- C Устанавливается, если произошел заем. В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий бит результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
SUBC Rw_n, Rw_m	30 nm	2
SUBC $Rw_n, [Rw_i]$	38 n:10ii	2
SUBC $Rw_n, [Rw_i+]$	38 n:11ii	2
SUBC $Rw_n, \#data_3$	38 n:0###	2
SUBC $reg, \#data_{16}$	36 RR ## ##	4
SUBC reg, mem	32 RR MM MM	4
SUBC mem, reg	34 RR MM MM	4

SUBCB Целочисленное вычитание с переносом

Синтаксис SUBCB op1, op2

Действие $(op1) \leftarrow (op1) - (op2) - (C)$

Тип данных BYTE

Описание Выполняется вычитание содержимого источника op2 и бита переноса C из содержимого приемника op1. Результат сохраняется в op1. Эта инструкция используется для реализации вычислений с повышенной точностью.

Флаги состояния

E	Z	V	C	N
*	S	*	S	*

- E Устанавливается, если значение op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю и перед выполнением команды был установлен флаг Z. В противном случае сбрасывается.
- V Устанавливается, если произошло переполнение, т.е. результат не может быть представлен указанным типом данных. В противном случае сбрасывается.
- C Устанавливается, если произошел заем. В противном случае сбрасывается.
- N Устанавливается, если установлен старший значащий бит результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
SUBCB Rb _n , Rb _m	31 nm	2
SUBCB Rb _n , [Rw _i]	39 n:10ii	2
SUBCB Rb _n , [Rw _i +]	39 n:11ii	2
SUBCB Rb _n , #data ₃	39 n:0###	2
SUBCB reg, #data ₈	37 RR ## xx	4
SUBCB reg, mem	33 RR MM MM	4
SUBCB mem, reg	35 RR MM MM	4

TRAP Программное прерывание

Синтаксис TRAP op1

Действие

```
(SP) ← (SP) - 2
((SP)) ← (PSW)
IF (SYSCON.SGTDIS = 0) THEN
    (SP) ← (SP) - 2
    ((SP)) ← (CSP)
    (CSP) ← 0
END IF
(SP) ← (SP) - 2
((SP)) ← (IP)
(IP) ← op1 * 4
```

Описание Производится обработка программного прерывания: на вершине системного стека сохраняется содержимое регистров PSW, CSP и IP и производится переход по адресу вектора прерывания, определяемому операндом op1. Перед каждым сохранением значение указателя стека SP уменьшается на 2. Содержимое CSP сохраняется, если разрешена сегментация (бит SGTDIS в регистре SYSCON равен нулю). Переход на обработчик по адресу вектора прерывания осуществляется аппаратно после прихода запроса или программно с помощью инструкции TRAP. Обработчику не передается никакой дополнительной информации о том, какой тип перехода произошел. Сохранение PSW, CSP и IP производится аналогично аппаратному переходу. В отличие от аппаратного прерывания, уровень приоритета ЦПУ не изменяется. Для возврата из обработчика следует использовать инструкцию RETI.

Флаги состояния

E	Z	V	C	N
-	-	-	-	-

E	Не изменяется.
Z	Не изменяется.
V	Не изменяется.
C	Не изменяется.
N	Не изменяется.

Формат инструкции

Мнемоника	Формат	Размер
TRAP #trap ₇	9B t:ttt0	2

XOR Исключающее ИЛИ

Синтаксис XOR op1, op2

Действие $(op1) \leftarrow (op1) \oplus (op2)$

Тип данных WORD

Описание Выполняется операция побитового исключающего ИЛИ над содержимым операнда op2 и содержимым приемника op1. Результат сохраняется в op1.

Флаги состояния

E	Z	V	C	N
*	*	0	0	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу. В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю.
В противном случае сбрасывается.
- V Всегда сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
XOR Rw_n, Rw_m	50 nm	2
XOR $Rw_n, [Rw_1]$	58 n:10ii	2
XOR $Rw_n, [Rw_1+]$	58 n:11ii	2
XOR $Rw_n, \#data_3$	58 n:0###	2
XOR $reg, \#data_{16}$	56 RR ## ##	4
XOR reg, mem	52 RR MM MM	4
XOR mem, reg	54 RR MM MM	4

XORB Исключающее ИЛИ

Синтаксис XORB op1, op2

Действие (op1) ← (op1) ⊕ (op2)

Тип данных BYTE

Описание Выполняется операция побитового исключающего ИЛИ над содержимым операнда op2 и содержимым приемника op1. Результат сохраняется в op1.

Флаги состояния

E	Z	V	C	N
*	*	0	0	*

- E Устанавливается, если содержимое op2 равно наименьшему отрицательному числу.
В противном случае сбрасывается.
- Z Устанавливается, если результат равен нулю.
В противном случае сбрасывается.
- V Всегда сбрасывается.
- C Всегда сбрасывается.
- N Устанавливается, если установлен старший значащий разряд результата. В противном случае сбрасывается.

Формат инструкции

Мнемоника	Формат	Размер
XORB Rb _n , Rb _m	51 nm	2
XORB Rb _n , [Rw _i]	59 n:10ii	2
XORB Rb _n , [Rw _i +]	59 n:11ii	2
XORB Rb _n , #data ₃	59 n:0###	2
XORB reg, #data ₈	57 RR ## xx	4
XORB reg, mem	53 RR MM MM	4
XORB mem, reg	55 RR MM MM	4

Содержание

Обозначения	1
ADD Целочисленное сложение	6
ADDB Целочисленное сложение	7
ADDC Целочисленное сложение с переносом	8
ADDCB Целочисленное сложение с переносом	9
AND Логическое И	10
ANDB Логическое И	11
ASHR Арифметический сдвиг в сторону младших адресов	12
ATOMIC Временное запрещение обработки запросов на прерывание	13
BAND Битовое логическое И	14
BCLR Очистка бита	15
BCMP Сравнение битов	16
BFLDH Битовое поле старшего байта	17
BFLDL Битовое поле младшего байта	18
BMOV Битовая пересылка	19
BMOVB Битовая пересылка с инверсией	20
BOR Битовое ИЛИ	21
BSET Установка бита	22
BXOR Битовое исключающее ИЛИ	23
CALLA Абсолютный вызов подпрограммы	24
CALLI Косвенный вызов подпрограммы	25
CALLR Относительный вызов подпрограммы	26
CALLS Межсегментный вызов подпрограммы	27
CMP Целочисленное сравнение	28
CMPB Целочисленное сравнение	29
CMPD1 Целочисленное сравнение с уменьшением на единицу	30
CMPD2 Целочисленное сравнение с уменьшением на 2	31
CMPI1 Целочисленное сравнение с увеличением на единицу	32
CMPI2 Целочисленное сравнение с увеличением на 2	33
CPL Поразрядная инверсия	34
CPLB Поразрядная инверсия	35
DISWDT Запрещение работы сторожевого таймера	36
DIV Деление со знаком(16 разрядов / 16 разрядов)	37
DIVL Деление со знаком(32 разряда / 16 разрядов)	38
DIVLU Беззнаковое деление(32 разряда / 16 разрядов)	39
DIVU Беззнаковое деление(16 разрядов / 16 разрядов)	40
EINIT Конец инициализации	41
EXTR Переадресация регистров	42
EXTP Страничная переадресация	44
EXTPR Страничная переадресация и переадресация регистров	46
EXTS Сегментная переадресация	48
EXTSR Сегментная переадресация и переадресация регистров	50
IDLE Режим ожидания	52
JB Относительный переход, если бит установлен	53

JBC	Относительный переход с очисткой бита, если бит установлен ..	54
JMPA	Абсолютный условный переход	55
JMPI	Косвенный условный переход	56
JMPR	Относительный условный переход	57
JMPS	Абсолютный межсегментный переход	58
JNB	Относительный переход, если бит сброшен	59
JNBS	Относительный переход с установкой бита, если бит сброшен...	60
MOV	Пересылка данных	61
MOVB	Пересылка данных	62
MOVBS	Пересылка байта с расширением знака	63
MOVBSZ	Пересылка байта с очисткой старшего байта	64
MUL	Умножение со знаком	65
MULU	Беззнаковое умножение	66
NEG	Инверсия знака	67
NEGB	Инверсия знака	68
NOP	Нет операции	69
OR	Логическое ИЛИ	70
ORB	Логическое ИЛИ	71
PCALL	Абсолютный вызов подпрограмм с сохранением слова на стеке	72
POP	Извлечь слово из системного стека	73
PRIOR	Счетчик нормализации	74
PUSH	Поместить слово на стек	75
PWRDN	Режим пониженного потребления	76
RET	Возврат из подпрограммы	77
RETI	Возврат из подпрограммы обработки прерывания	78
RETP	Возврат из подпрограммы извлечение слова	79
RETS	Межсегментный возврат из подпрограммы	80
ROL	Циклический сдвиг в сторону старших разрядов	81
ROR	Циклический сдвиг в сторону младших разрядов	82
SCXT	Переключение контекста	83
SHL	Сдвиг в сторону старших адресов	84
SHR	Сдвиг в сторону младших адресов	85
SRST	Программный сброс	86
SRVWDT	Перезапуск сторожевого таймера	87
SUB	Целочисленное вычитание	88
SUBB	Целочисленное вычитание	89
SUBC	Целочисленное вычитание с переносом	90
SUBCB	Целочисленное вычитание с переносом	91
TRAP	Программное прерывание	92
XOR	Исключающее ИЛИ	93
XORB	Исключающее ИЛИ	94

Для заметок