

МЕТОДЫ АДРЕСАЦИИ ПРОЦЕССОРА K1801BM2

Обозначение	Код	Метод	Время обращения*
R Операнд в регистре	0R	Регистровый	1
(R) В регистре адрес	1R	Косвенно-регистровый	1.3
(R)+ В регистре адрес операнда. Содержимое регистра после его использования увеличивается на 1 или 2	2R	Автоинкрементный	1.3
@(R)+ В регистре адрес адреса операнда. Содержимое регистра после его использования увеличивается на 1 или 2	3R	Косвенно-автоинкрементный	3.3
-(R) В регистре адрес операнда. Содержимое регистра до его использования уменьшается на 1 или 2.	4R	Автодекрементный	
@-(R) В регистре адрес адреса операнда. Содержимое регистра до его использования уменьшается на 1 или 2.	5R	Косвенно-автодекрементный	3.3
E(R) Адрес операнда — сумма содержимого регистра и индексного слова.	6R	Индексный	
@E(R) Адрес адреса операнда — сумма содержимого регистра и индексного слова.	7R	Косвенно-индексный	4
#E Число E является операндом	27	Непосредственный	2.5
@#E Число E является абсолютным адресом операнда	37	Абсолютный	3.3
E Адрес операнда — сумма индексного слова и СК	67	Относительный	3.3
@E Адрес адреса операнда — сумма индексного слова и СК.	77	Косвенно-относительный	4

* Измерено для команды TST в условных единицах (1 единица соответствует времени выполнения команды TST RN).

СИСТЕМА КОМАНД ПРОЦЕССОРА K1801BM2

Мнемоника	Код	Операция	Алгоритм	H	P	T	N	Z	V	C
MOV(B)	.1SSDD	Пересылка *(1)	d <- s	-	-	-	.	.	0	-
CMP(B)	.2SSDD	Сравнение	s-d	-	-	-
ADD	06SSDD	Сложение	d <- s+d	-	-	-
SUB	16SSDD	Вычитание	d <- s-d	-	-	-
BIT(B)	.2SSDD	Проверка битов	s and d	-	-	-	.	.	0	-
BIC(B)	.4SSDD	Очистка битов	d <- (not s) and d	-	-	-	.	.	0	-
BIS(B)	.5SSDD	Установка битов	d <- s or d	-	-	-	.	.	0	-
XOR	074RDD	Исключающее ИЛИ	d <- R xor d	-	-	-	.	.	0	-
CLR(B)	.050DD	Очистка	d <- 0	-	-	-	0	1	0	0
COM(B)	.051DD	Инвертирование	d <- not d	-	-	-	.	.	0	1
INC(B)	.052DD	Прибавление единицы	d <- d+1	-	-	-	.	.	.	-
DEC(B)	.053DD	Вычитание единицы	d <- d-1	-	-	-	.	.	.	-
NEG(B)	.054DD	Изменение знака	d <- -d	-	-	-

TST(B)	.057DD	Проверка		- - - . . 0 0
ROR(B)	.060DD	Циклический сдвиг вправо	-> C,d	- - -
ROL(B)	.061DD	Циклический сдвиг влево	d,C <-	- - -
ASR(B)	.062DD	Арифметический сдвиг вправо	d/2	- - -
ASL(B)	.063DD	Арифметический сдвиг влево	2d	- - -
SWAB	0003DD	Перестановка байтов		- - - . . . 0
ADC(B)	055DD	Прибавление переноса	d+C	- - -
SBC(B)	056DD	Вычитание переноса	d-C	- - -
SXT	0067DD	Расширение знака	d <- 0 or -1	- - - - . 0 -
NOP	000240	Нет операции		- - - - - - - -
CLC	000241	Очистка бита C в ССП *(2)	C <- 0	- - - - - - - 0
CLV	000242	Очистка бита V в ССП	V <- 0	- - - - - 0 -
CLZ	000244	Очистка бита Z в ССП	Z <- 0	- - - - 0 - -
CLN	000250	Очистка бита N в ССП	N <- 0	- - - 0 - - -
CCC	000257	Очистка битов NZVC в ССП	NZVC <- 0	- - - 0 0 0 0
SEC	000261	Установка бита C в ССП	C <- 1	- - - - - - - 1
SEV	000262	Установка бита V в ССП	V <- 1	- - - - - 1 -
SEZ	000264	Установка бита Z в ССП	Z <- 1	- - - - 1 - -
SEN	000270	Установка бита N в ССП	N <- 1	- - - 1 - - -
SCC	000277	Установка битов NZVC в ССП	NZVC <- 1	- - - 1 1 1 1
MUL	070RSS	Умножение *(3)	RnRn+1 <- Rn*s	- - - . . 0 .
DIV	071RSS	Деление *(4)	RnRn+1 <- RnRn+1/s	- - -
ASH	072RSS	Арифметический сдвиг	R <- R*/2^s -	- - -
ASHC	073RSS	Арифметический сдвиг двойного слова RnRn+1 <- RnRn+1*/2^s		- - -
FADD	07500R	Сложение с плавающей запятой *(5)		- - - * * 0 0
FSUB	07501R	Вычитание с плавающей запятой		- - - * * 0 0
FMUL	07502R	Умножение с плавающей запятой		- - - * * 0 0
FDIV	07503R	Деление с плавающей запятой		- - - * * 0 0
MFPS	1067DD	Чтение ССП	d <- ССП	- - - . . 0 -
MTPS	1064DD	Запись ССП	ССП <- d	- . -
BR	0004XXX	Безусловное ветвление *(6)		
BNE	0010XXX	Ветвление, если не равно (нулю)		[Z=0]
BEQ	0014XXX	Ветвление, если равно (нулю)		[Z=1]
BPL	1000XXX	Ветвление, если плюс		[N=0]
BMI	1004XXX	Ветвление, если минус		[N=1]
BVC	1020XXX	Ветвление, если нет арифметического переполнения		[V=0]
BVS	1024XXX	Ветвление, если есть арифметическое переполнение		[V=1]
BCC	1030XXX	Ветвление, если нет переноса		[C=0]
BCS	1034XXX	Ветвление, если есть перенос		[C=1]
BGE	0020XXX	Ветвление, если больше или равно (нулю)		[N xor C=0]
BLT	0024XXX	Ветвление, если меньше (нуля)		[N xor C=1]
BGT	0030XXX	Ветвление, если больше (нуля)		[Z and (NxorV)=0]
BLE	0034XXX	Ветвление, если меньше или равно (нулю)		[Z and (NxorV)=1]

BHI	1010XXX	Ветвление, если больше	[C and Z=0]
BLOS	1014XXX	Ветвление, если меньше или равно	[C and Z=1]
BHIS	1030XXX	См. BCC (введена для удобства программиста)	
BLO	1034XXX	См. BCS (введена для удобства программиста)	
JMP	0001DD	Безусловный переход	
JSR	004RDD	Обращение к подпрограмме	
RTS	00020R	Возврат из подпрограммы	
MARK	0064NN	Восстановление указателя стека	
SOB	077RNN	Вычитание единицы и ветвление в обратном направлении	
EMT	104000— 104377	Командное прерывание для системных программ	
TRAP	104400— 104777	Командное прерывание для программ пользователя	0
			0
IOT	000004	Командное прерывание для ввода - вывода	0
			0
BPT	000003	Командное прерывание для отладки	0
			0
RTI	000002	Возврат из прерывания *(7)	- . -
		СК <- (SP)+, ССП <- (SP)+	*
RTT	000006	Возврат из прерывания *(7)	
		СК <- (SP)+, ССП <- (SP)+	*
WAIT	000001	Ожидание прерывания	- - - - -
RESET	000005	Сброс внешних устройств	- - - - -
HALT	000000	Останов процессора. Осуществляется принудительная установка режима «HALT», СК и РСР загружаются из ВП SEL- ОЗУ, КРСК <- СК, КРСР <- РСР, СК <- (160170), РСР <- (160172)	
Дополнительные команды режима HALT (в режиме «USER» — резервные команды):			
ПУСК	10-13	Если нет запросов на прерывание, то осуществляется переход в режим «USER» и начинается выполняться программа с адреса в КРСК	СК <- КРСК, РСР <- КРСР
ШАГ	14-17	В отличие от ПУСК, управление безусловно передается команде по адресу в КРСК, после выполнения которой осуществляется возврат в режим HALT к следующей за «ШАГ» инструкции	СК <- КРСК, РСР <- КРСР
ЧПТ	20, 30	R0 загружается значением внешнего регистра по процедуре безадресного чтения	
		R0 <- (SEL)	- - - - -
ЧЧП	21	R0 загружается значением ячейки памяти режима «USER» (в том числе и из страницы ввода-вывода) по адресу в R5. Соответствует команде MOV (R5)+, R0 в режиме «USER»	
		R0 <- (R5)+	- - - - -
ЗЧП	31	Значение R0 пересылается в ячейку ОЗУ режима «USER» (в том числе и в регистр страницы ввода-вывода) по адресу в R5. Соответствует команде MOV R0, -(R5) в режиме «USER»	
ЧКСК	22, 23	-(R5)<-R0	- - - - -
		R0 загружается R0 <- КРСК	- - - - -

ЧКСП	24-27	значением КРСК R0 загружается	R0 <- КРСП	- - - - -
ЗКСК	32, 33	значением КРСП Значение R0	КРСК <- R0	- - - - -
ЗКСП	34-37	загружается в КРС Значение R0	КРСП <- R0	- - - - -
		загружается в КРСП		

- *(1) Команда MOVВ (единственная среди байтовых команд) с использованием регистра в качестве приемника расширяет старший разряд младшего байта (очищает или устанавливает все разряды старшего байта в зависимости от знака младшего байта).
- *(2) Сочетания команд изменения признаков, соединенные по схеме «ИЛИ» образуют комбинированные команды.
- *(3) По команде MUL перемножаются операнды источника и приемника, взятые в двоичном дополнительном коде. Если регистр приемника имеет четный номер, 32-разрядный результат помещается в Rn и Rn+1, иначе в регистре Rn сохраняется только младшая часть результата.
- *(4) 32-разрядное слово в двоичном дополнительном коде делится на операнд источника. Частное заносится в Rn, а остаток в Rn+1. Знак остатка совпадает со знаком делимого. Команда работает только с четным регистром.
- *(5) Команды арифметики с плавающей запятой обрабатываются в режиме "HALT" подпрограммой обработки прерываний по вектору 160010.
- *(6) Команды ветвления ССП не изменяют.
- *(7) После выполнения команды RTT проверка T разряда ССП не производится (он проверяется начиная со следующей за RTT командой), после выполнения команды RTI разряд T начинает проверяться сразу.

Биты слова состояния процессора:

8 7 4 3 2 1 0

Н Р Т N Z V C

C — перенос;

T — прерывание по разряду T;

V — арифметическое
переполнение;

P — приоритет;

Z — нулевой результат;

N — режим HALT / USER.

N — отрицательный результат;

Обозначения поля «Код»:

SS — 6 разрядов адресации
источника;

XXX — 8 разрядов смещения (со
знаком);

DD — 6 разрядов адресации
приемника;

NN — 6 разрядов смещения (без
знака);

. — бит 0 или 1 в соответствии со словной или байтовой
операцией.

Обозначения действий (поле «Алгоритм»):

s — содержимое источника;

and — логическая операция «И»;

d — содержимое приемника;

or — логическая операция «ИЛИ»;

R — содержимое регистра;

xor — логическая операция
«исключающее ИЛИ».

<- — направление пересылки;

Обозначения изменения битов слова состояния процессора (поле «HPTNZVC»):

* — в зависимости от обстоятельств очищается или устанавливается;

. — изменяется;

0 — очищается;

- — не изменяется;

1 — устанавливается.