

---

Однокристалльный микропроцессор  
K1801BM1

**Техническое описание**

**Версия 1.2 (January 4, 2020)**

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 1. Введение

Данный документ описывает микросхему советского однокристалльного микропроцессора K1801BM1. В открытом доступе не имеется официального технического описания данного процессора, а различные источники не являются полными и непротиворечивыми, поэтому было принято решение составить техническое описание, основанное на имеющихся фрагментах официальных документов, проверенных на практике фактов и результатах реверс-инжиниринга собственно микросхемы K1801BM1.

По возможности ссылки на использованные материалы приведены в соответствующем разделе.

В тексте не делается различия между разными исполнениями корпуса - KM1801BM1 (керамический), KP1801BM1 (пластиковый корпус) и прочими, вместо этого употребляется единое обозначение K1801BM1.

Изначально, когда выполнялся реверс-инжиниринг микросхемы K1801BM1A, предполагалось что микросхемы с другими буквенными индексами (K1801BM1Б, K1801BM1В, K1801BM1Г) являются результатом сортировки по частоте и работоспособности отдельных блоков после выполнения заводских тестов. Данное предположение оказалось только частично верным. Микросхема K1801BM1Г выпускалась по другим фотошаблонам слоя диффузии и содержит микропрограмму, отличную от микропрограммы микросхемы K1801BM1А. Функциональные отличия микросхемы K1801BM1Г заключаются в поддержке команды умножения, в гарантированно работоспособном блоке таймера и поддержке прерывания от этого блока. Поскольку данные отличия незначительны, то возможно написание единой документации на все варианты K1801BM1, с акцентированным описанием дополнительных особенностей версии K1801BM1Г.

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 2. Назначение

Микросхема K1801BM1 является однокристалльным 16-ти разрядным микропроцессором (далее по тексту процессор), предназначенным для обработки цифровой информации.

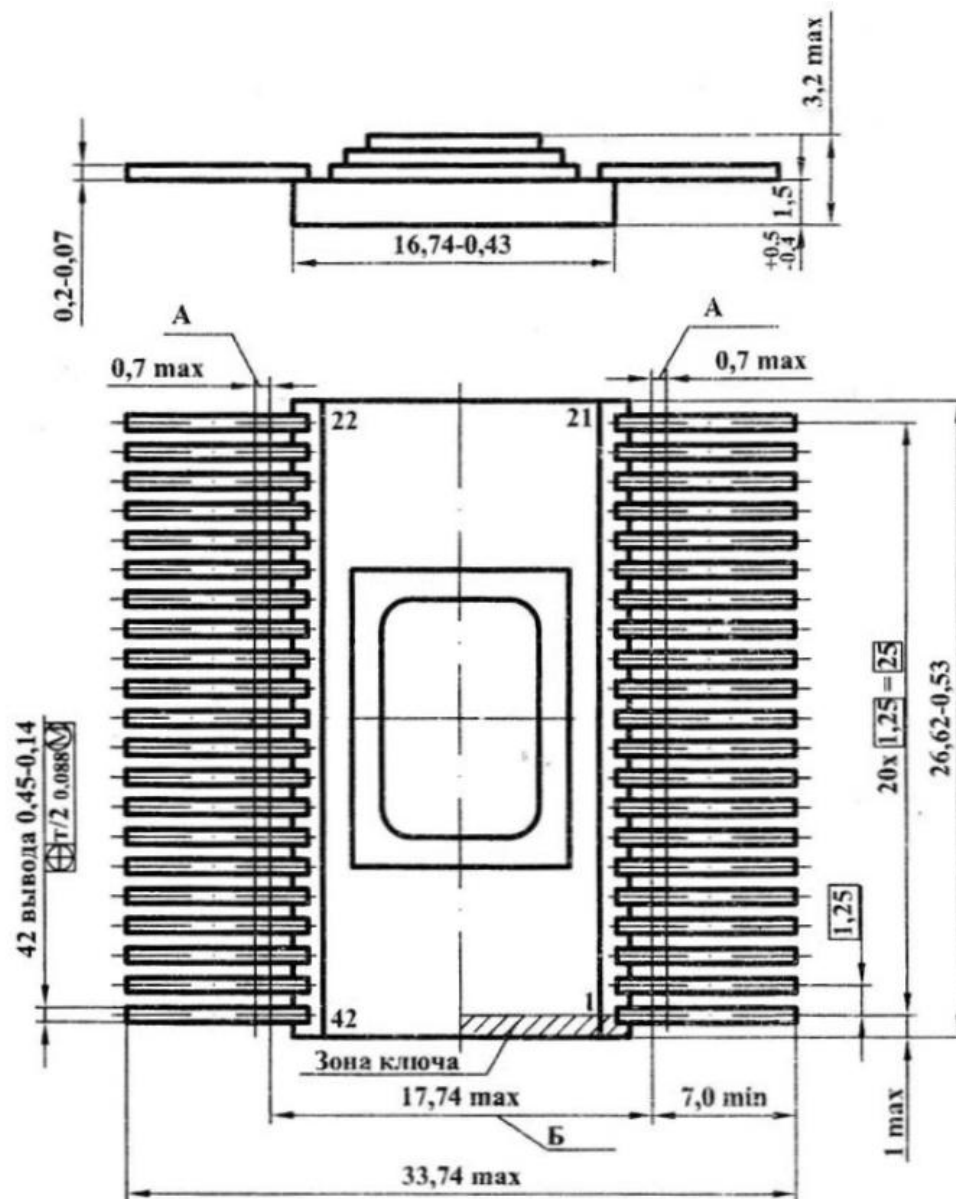
В составе ЭВМ процессор может использоваться для управления технологическими процессами, в контрольно-измерительной аппаратуре, в системах связи, для решения инженерно-технических и экономических задач.

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

### 3. Технические характеристики

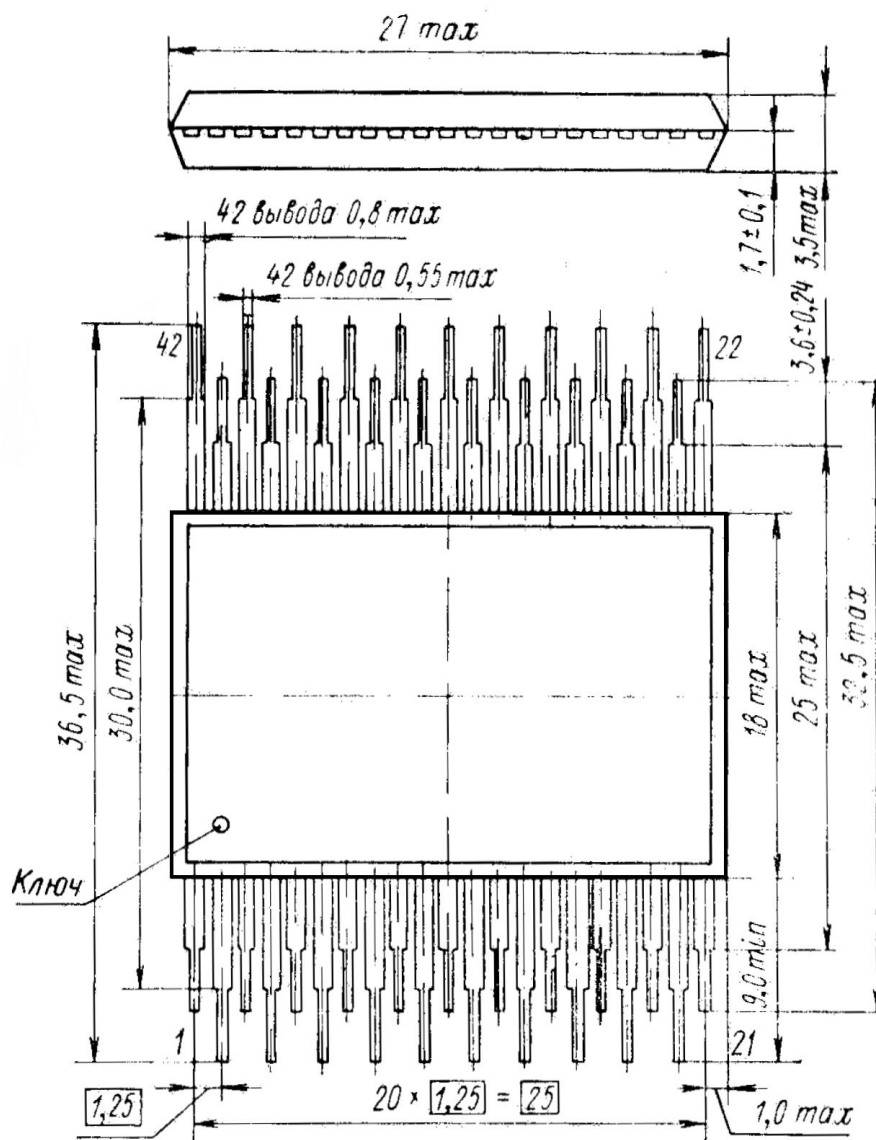
Представление чисел	дополнительный код с фиксированной запятой
Система команд	безадресная, одноадресная, двухадресная
Виды адресации	регистровая, косвенно-регистровая, автоинкрементная, косвенно-автоинкрементная, автодекрементная, косвенно-автодекрементная, индексная, косвенно-индексная
Число регистров общего назначения	8
Число уровней прерываний	4
Системная магистраль	типа МПИ
Адресное пространство	64 килобайта
Тактовая частота K1801BM1A K1801BM1B K1801BM1B K1801BM1Г	минимальная 100 кГц для всех вариантов (ограничение частоты снизу связано с тем, что для построения некоторых защелок используется емкость затворов n-МОП транзисторов)  до 4.7 МГц (BM1A) до 3.5 МГц (BM1B) до 2.5 МГц (BM1B) до 4.7 МГц (BM1Г)
Максимальное быстродействие (регистровые операции)	до 500 тысяч операций в секунду
Напряжение питания	однополярное единое питание +5 В ±10%
Многопроцессорность	до 4-х процессоров на единой магистрали
Уровни входных логических сигналов низкого уровня высокого уровня	не более 0.6 В, при токе не утечки не более 50мкА не менее 2.4 В, при токе не утечки не более 50мкА
Уровни выходных логических сигналов низкого уровня высокого уровня	не более 0.5 В, при втекающем токе 3.2мА не менее 2.4 В, при вытекающем токе -0.2 мА
Емкость нагрузки	до 100 пФ
Технология изготовления	n-МОП, типовая ширина затвора транзистора ~3 мкм, один слой металла и один слой поликремния
Размер кристалла	5x5 мм
Конструкция	Металлокерамический корпус типа 429.42 или пластиковый 2204.42-3
Количество транзисторов	~16650 штук
Потребляемая мощность	до 1.2 Вт

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	



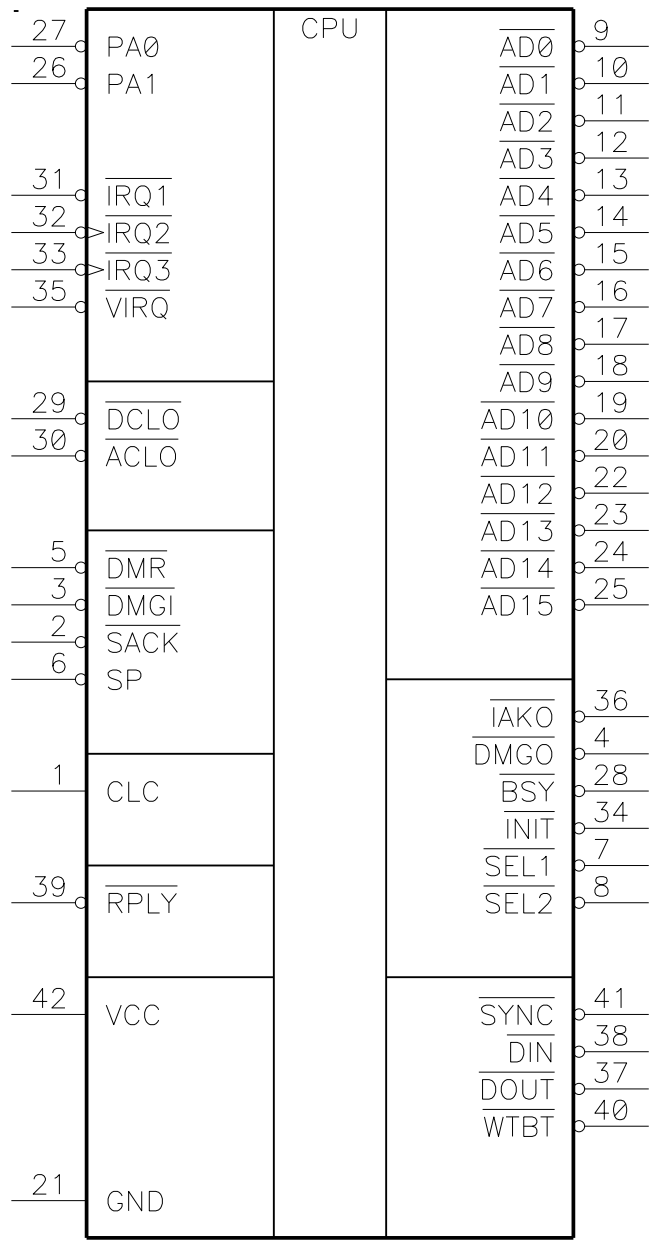
Чертеж планарного металлокерамического корпуса типа 429.42-5

Однокристальный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	



Чертеж планарного пластикового корпуса 2204.42-3

Однокристальный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	



Условное графическое обозначение K1801BM1

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

#### 4. Назначение выводов

1	<b>CLC</b>	вход	Основной тактовый сигнал частотой не менее 100 кГц, ограничение частоты снизу связано с широким применением динамической логики в схеме процессора. Под динамической логикой понимается использование емкостных свойств затворов n-МОП транзисторов, вследствие чего транзисторы могут некоторое время сохранять свое состояние при затворе, отключенном от источника сигнала
2	<b>nSACK</b>	вход выход-ОК	Подтверждение захвата магистрали внешним агентом  Ведущий процессор с аппаратным номером 0 никогда не активизирует данный выход (типа открытый коллектор). Ведомые процессоры с номерами 1-3 выставляют на данном выходе низкий уровень в процессе обмена по магистрали.  Как вход используется всеми процессорами для общего контроля занятости магистрали
3	<b>nDMGI</b>	вход	Разрешение от внешнего арбитража на доступ к магистрали, если процессор в данный момент занят выполнением внутренних операций и ему не требуется доступ к магистрали, то обеспечивается дальнейшее прохождение разрешающего сигнала на выход <b>nDMGO</b>
4	<b>nDMGO</b>	выход	Разрешение на доступ к магистрали другим, менее приоритетным модулям. Активный низкий уровень появляется в моменты когда на входе <b>nDMGI</b> низкий уровень и внутреннему процессорному блоку не требуется доступ к магистрали
5	<b>nDMR</b>	вход выход-ОК	Для ведущего процессора с аппаратным номером 0 этот вывод является входом запроса на доступ к магистрали от ведомых процессоров с номерами 1-3 и других внешних модулей системы, ведущий процессор никогда не использует данный вывод как выход.  Ведомые процессоры с номерами 1-3 используют данный вывод как выход запроса на доступ к магистрали
6	<b>nSP</b>	вход	Вход периферийного таймера, может использоваться как внешняя тактовая частота, не превышающая одной восьмой тактовой частоты процессора, или как вход защелкивания текущего значения счетчика таймера в отдельном регистре. Все действия в таймере (счет и защелкивание) происходят по ниспадающему фронту сигнала на данном входе с задержкой от 1 до 8 тактов процессорной частоты <b>CLC</b>
7	<b>nSEL1</b>	выход-ОК	Сигнал подтверждения обращения по адресу база внутреннего периферийного блока + 16 <sub>8</sub> . Формируется всегда, независимо от того какой агент, внутренний или внешний обращается к данному адресу.
8	<b>nSEL2</b>	выход-ОК	Сигнал подтверждения обращения по адресу база внутреннего периферийного блока + 14 <sub>8</sub> . Формируется всегда, независимо от того какой агент, внутренний или внешний обращается к данному адресу.
9-20, 22-25	<b>nAD0- nAD15</b>	входы выходы-3	Линии 16-ти разрядной мультиплексированной шины адреса и данных. Сигналы передаются по данным линиям в инвертированном



Однокристальный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

			<p>виде. Передача значения адреса и данных осуществляется методом разделения во времени.</p> <p>Если системной магистралью МПИ владеет внутренний процессорный блок, то выходы <b>nAD</b> работают как выходы:</p> <ul style="list-style-type: none"> <li>- в фазе передачи адреса</li> <li>- в фазе записи данных во всем диапазоне адресов</li> <li>- в фазе чтения данных из диапазона адресов внутреннего периферийного блока, за исключением двух старших адресов <b>nSEL1</b> и <b>nSEL2</b>.</li> </ul> <p>Если системной магистралью МПИ владеет внешний по отношению к микросхеме агент, то выходы <b>nAD</b> работают как выходы:</p> <ul style="list-style-type: none"> <li>- в фазе чтения данных по адресам внутреннего периферийного блока</li> </ul> <p>В остальное время выходы находятся в высокоимпедансном состоянии. Входы в целях мониторинга адреса транзакции и данных записи работают постоянно.</p>
26, 27	<b>nPA1, nPA0</b>	входы	<p>Аппаратный номер процессора в многопроцессорной системе, сигналы на данных входах являются инвертированными.</p> <p><b>nPA1=1, nPA0=1</b> - процессор номер 0, высший приоритет  <b>nPA1=1, nPA0=0</b> - процессор номер 1  <b>nPA1=0, nPA0=1</b> - процессор номер 2  <b>nPA1=0, nPA0=0</b> - процессор номер 3, низший приоритет</p> <p>Если процессор единственный в системе (однопроцессорный вариант), то на данные входы должен быть подан высокий уровень.</p> <p>От аппаратного номера процессора зависит базовый адрес блока регистров внутреннего периферийного блока.</p>
28	<b>nBSY</b>	выход-ОК	Низкий уровень на данной выходе сигнализирует о выполнении транзакции на шине МПИ
29	<b>nDCLO</b>	вход	Авария источника питания постоянного напряжения. Низкий уровень на данном входе выставляется внешней схемой при выходе питающего напряжения системы (обычно +5V и +12V) за допустимые рабочие диапазоны, при этом происходит безусловный аппаратный сброс процессора.
30	<b>nACLO</b>	вход	Авария сетевого источника питания. Низкий уровень на данном входе выставляется внешней схемой при выходе сетевого питающего напряжения за пределы рабочего диапазона, при этом внутренние постоянные питающие напряжения все еще могут обеспечиваться. Ниспадающий фронт вызывает прерывание по вектору 024 <sub>8</sub> , для устойчивой работы процессора сигнал следует синхронизировать по нарастающему фронту с тактовой частоты <b>CLC</b> . По нарастающему фронту сигнала происходит старт микропрограммного автомата, если он был сброшен ранее активным сигналом <b>nDCLO</b>
31	<b>nIRQ1</b>	вход	Первый запрос на радиальное прерывание, низкий уровень вызывает прерывание по вектору 160002 <sub>8</sub> . Для устойчивой работы процессора сигнал следует синхронизировать по нарастающему фронту с

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

			тактовой частоты <b>CLC</b>
32	<b>nIRQ2</b>	вход	Второй запрос на радиальное прерывание, ниспадающий фронт вызывает прерывание по вектору 100 <sub>8</sub> , для устойчивой работы процессора сигнал следует синхронизировать по нарастающему фронту с тактовой частоты <b>CLC</b>
33	<b>nIRQ3</b>	вход	Третий запрос на радиальное прерывание, ниспадающий фронт вызывает прерывание по вектору 270 <sub>8</sub> , для устойчивой работы процессора сигнал следует синхронизировать по нарастающему фронту с тактовой частоты <b>CLC</b>
34	<b>nINIT</b>	вход выход-ОК	Вход сброса периферийных устройств. Низкий уровень на данном входе переводит периферийный таймер в исходное состояние и временно снимается активный запрос <b>nDMR</b> . Также приводятся в исходное состояние детекторы ниспадающего фронта на входах <b>nIRQ2</b> и <b>nIRQ3</b> .  На выходе (открытый коллектор) формируется активный сигнал низкого уровня при выполнении инструкции <b>RESET</b> и/или аппаратном сбросе процессора
35	<b>nVIRQ</b>	вход	Запрос на векторное прерывание, низкий уровень вызывает цикл подтверждения прерывания на шине МПИ и переход по прочитанному из внешнего устройства вектору. Для устойчивой работы процессора сигнал следует синхронизировать по нарастающему фронту тактовой частоты <b>CLC</b> . Следует отметить, что только ведущий процессор с аппаратным номером 0 может принимать и обрабатывать прерывания по данному входу, в ведомых процессор с аппаратными номерами 1-3 этот вход заблокирован.
36	<b>nIAKO</b>	выход-3	Выход предоставления прерывания, низкий уровень на данном выходе появляется в циклах чтения вектора прерывания из внешнего устройства. Когда доступ к шине МПИ процессору не предоставлен выход находится в высокоимпедансном состоянии
37	<b>nDOUT</b>	вход выход-3	Вывод данных, вход используется внутренним периферийным блоком для выполнения транзакций записи. Выходом управляет процессорный блок в случае предоставления ему доступа к шине МПИ. Когда доступ к шине процессору не предоставлен выход находится в высокоимпедансном состоянии
38	<b>nDIN</b>	вход выход-3	Ввод данных, вход используется внутренним периферийным блоком для выполнения транзакций чтения. Выходом управляет процессорный блок в случае предоставления ему доступа к шине МПИ, когда доступ к шине процессору не предоставлен выход находится в высокоимпедансном состоянии
39	<b>nRPLY</b>	вход выход-ОК	Подтверждение обмена данными, вход используется процессорным блоком при выполнении транзакций на шине МПИ. На выходе генерируется активный низкий уровень периферийным блоком в случае любого обращения в его диапазон адресов, включая регистры <b>nSEL1</b> и <b>nSEL2</b> . Из-за выявленных ошибок в схеме процессора сигнал на данном входе следует синхронизировать с тактовым

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

			сигналом <b>CLC</b> (с любым из фронтов) для обеспечения устойчивой работы
40	<b>nWTBT</b>	выход-3	Запись/байт, в фазе трансляции адреса данный выход индицирует низким уровнем что в данной транзакции будет осуществляться вывод данных. Для циклов чтение и чтение-модификация-запись на данном выходе присутствует высокий уровень. В фазе обмена данными низкий уровень свидетельствует о записи одного из байтов, а не целого слова. Какой именно байт записывается определяется состоянием младшей адресной линии в фазе передачи адреса. Когда доступ к шине процессору не предоставлен выход находится в высокоимпедансном состоянии
41	<b>nSYNC</b>	вход выход-3	Вход используется периферийным блоком для мониторинга адреса транзакции. Выходом управляет процессорный блок при выполнении транзакций на шине МПИ. При этом выход имеет особенность - высокий уровень генерируется ровно на один такт частоты <b>CLC</b> , после чего выход переводится в высокоимпедансное состояние, поэтому для нормальной работы линия должна быть подтянута резистором к напряжению питания.
21	<b>GND</b>	земля	опорный уровень 0В, питание
42	<b>VCC</b>	питание	+5В с допуском $\pm 10\%$ , ток потребления до 240мА

Микросхема K1801BM1 содержит внутри два основных блока - собственно блок процессора и периферийный блок. Оба блока подключены к единой магистрали типа МПИ, которая также выходит на внешние выводы. Блок процессора может работать только как ведущий агент МПИ, периферийный блок может работать только как ведомый агент МПИ. Доступ процессорного блока к управлению МПИ контролируется модулем арбитража. Периферийный блок представляет собой набор различных специальных регистров, для периферийного блока выделен адресный диапазон из 8 16-ти разрядных слов, базовый адрес которого определяется аппаратным номером процессора. Внешние выводы МПИ микросхемы K1801BM1 выполняют совмещенные функции для процессорного и периферийного блоков. Доступ к периферийному блоку может осуществляться как со стороны внутреннего процессорного блока, так и со стороны внешнего по отношению к микросхеме процессора агента шины МПИ. Таким образом, можно выделить основные четыре режима работы внешних выводов МПИ микросхемы K1801BM1:

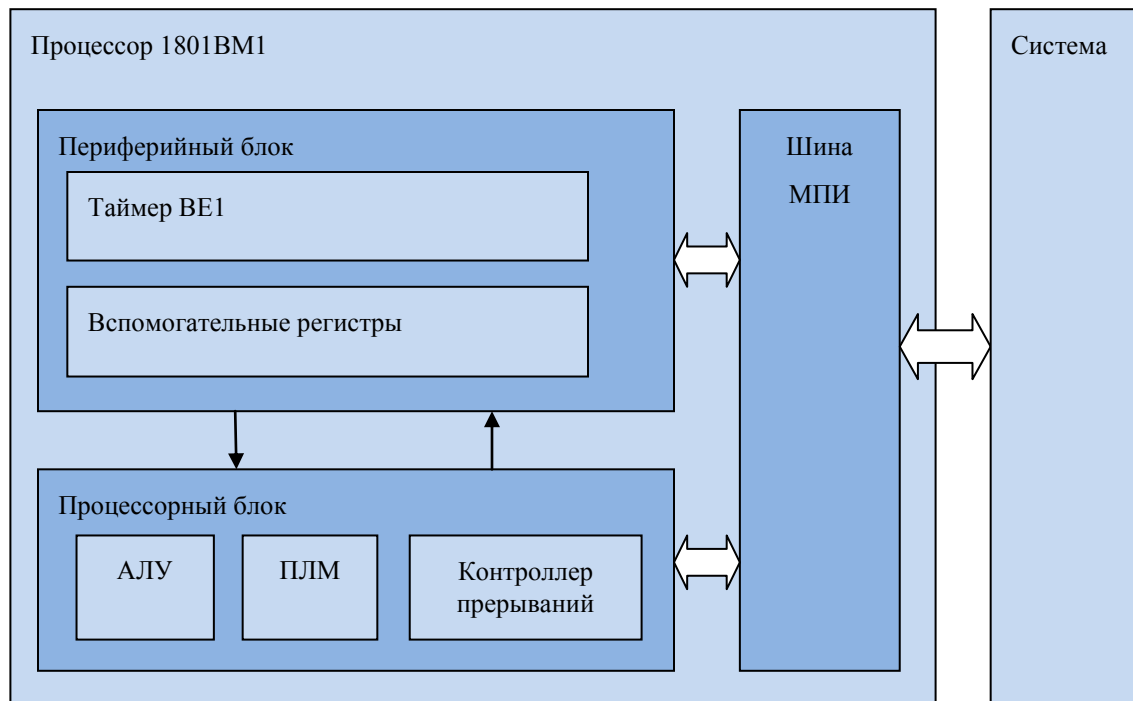
- доступ внутреннего процессорного блока к внешним адресам системы (которые, в том числе, могут быть адресами периферийного блока другой микросхемы K1801BM1 в многопроцессорной системе)
- доступ внутреннего процессорного блока к адресам внутреннего периферийного блока
- доступ внешнего агента к внешним адресам системы
- доступ внешнего агента к адресам внутреннего периферийного блока данного процессора

Особый случай представляют два старших адреса периферийного блока, эти регистры могут реализовываться на внешних схемах, но при этом управление обменом по МПИ осуществляется процессором, внешние схемы только представляют или получают данные, сигнал **nRPLY** формируется выходом K1801BM1 всегда, при обращении к любому адресу из диапазона периферийного блока.

Процессор K1801BM1 может работать в составе многопроцессорной системы с разделением общей магистрали МПИ. Система на основе K1801BM1 может содержать до 4-х процессоров. Процессорам

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

присваивается аппаратный номер от 0 до 3, в зависимости от уровней поданных на входы **nPA0-nPA1**. Процессор с аппаратным номером 0 называется ведущим и имеет высший приоритет при доступе к разделяемой шине МПИ, начиная цепочку арбитража по выводам **nDMGI/nDMGO**. Остальные процессоры с номерами 1-3 называются ведомыми, приоритет доступа к разделяемой шине определяется местом включения процессора в цепочку арбитража доступа к шине



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 5. Шина МПИ

Микропроцессор K1801BM1 для обмена с внешними устройствами использует системную шину типа МПИ (**М**агистральный **П**араллельный **И**нтерфейс). Обмен информацией между двумя устройствами по шине МПИ осуществляется по принципу ведомый-ведущий. Управление обменом осуществляет ведущее устройство, а ведомое устройство производит подтверждение приема или передачи слова данных. Некоторые устройства могут в разные моменты времени выполнять роль как ведущего так и ведомого. В каждый момент времени на шине МПИ может быть только одно ведущее устройство, которое физически осуществляет управление линиями магистрали. Доступ устройства к роли ведущего шины разрешается средствами приоритетного арбитража. Подробно с регламентом работы МПИ можно ознакомиться в стандарте "ГОСТ 26765.51-86 Интерфейс магистральный параллельный МПИ системы электронных модулей. Общие требования к совокупности правил обмена информацией" [6]. Данная же документация делает акцент именно на особенностях работы микропроцессора, для понимания принципов работы внешних устройств следует обратиться к упомянутому ГОСТ.

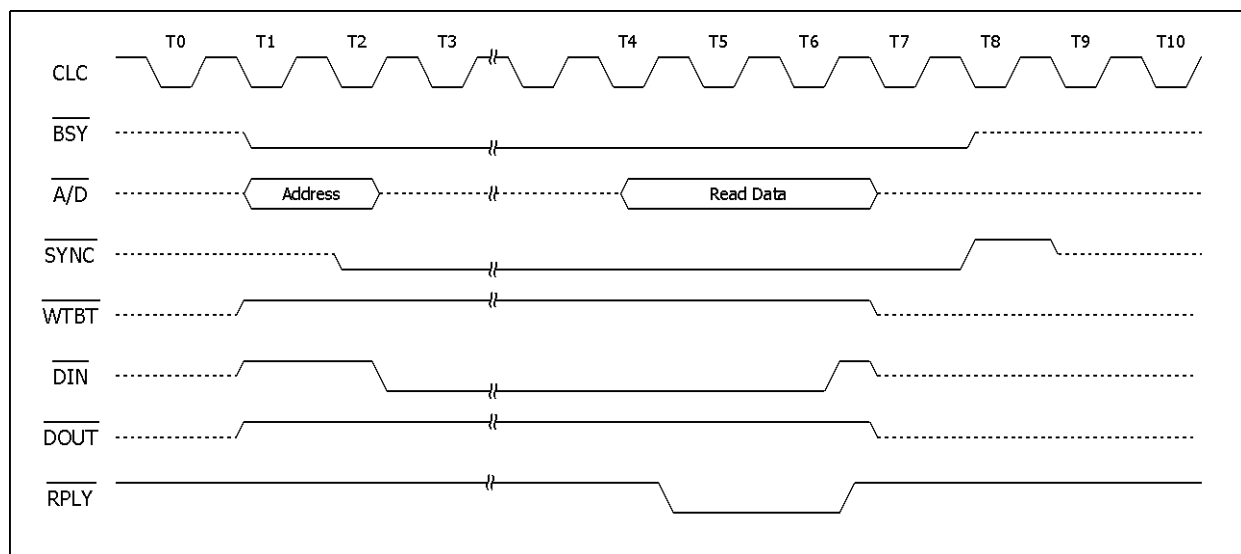
Микропроцессор K1801BM1 может выполнять роль ведущего устройства шины - в этом случае процессорный блок выполняет транзакции "Ввод" (чтение данных из внешнего устройства), "Вывод" (запись данных во внешнее устройство), "Ввод-Вывод" (чтение-модификация-запись), "Подтверждение прерывания" (чтение вектора прерывания). Так же микросхема K1801BM1 может выполнять роль ведомого устройства - в этом случае осуществляется доступ внешним агентом к периферийному блоку микросхемы, а также выполнять обе роли ведущего и ведомого устройств одновременно - при обращении процессорного блока к собственному (то есть, находящемуся в той же микросхеме) периферийному блоку. Блочные транзакции шины МПИ микропроцессор K1801BM1 не использует.

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 5.1 Транзакция "Чтение"

Ниже приведена диаграмма выполнения микропроцессором цикла чтения данных по шине МПИ в качестве ведущего устройства. Пунктирными линиями показано состояние высокого импеданса. Обычно в системе на линиях магистрали имеются подтягивающие резисторы, поэтому можно полагать что в эти моменты времени на линиях присутствует напряжение высокого уровня, все управляющие сигналы при этом принимают неактивное значение, на шине адрес/данных устанавливается логическое нулевое значения, поскольку шина инвертирована (напряжение низкого уровня соответствует логическому "1", напряжение высокого уровня соответствует логическому "0").

Также указана привязка установки и снятия управляющих выходов к фронтам тактового сигнала, при этом не учитывается внутренняя задержка буферных цепей, которая может достигать десятков наносекунд и, при высоком значении тактовой частоты, может быть сравнимой с длительностью такта.



В такте **T0** на магистрали МПИ нет ведущего устройства, обмен не выполняется, полагаем это состояние исходным. По срезу **T1** микропроцессор осуществляет захват магистрали (процесс арбитража рассмотрен подробно в соответствующем разделе), при этом разрешается управление выходами **nWTBT**, **nDIN**, **nDOUT**, **nIAKO**, **A/D**. Сигнал **nBSY** принимает активный низкий уровень, на шине адреса/данных устанавливается адрес читаемых данных. Выход сигнала **nSYNC** остается запрещенным - это особенность K1801BM1, для корректной работы шины в этом случае требуется внешний подтягивающий резистор. На выходе **nWTBT** установлен высокий уровень, что означает выполнение операции чтения. По срезу **T2** на выходе **nSYNC** формируется активный низкий уровень. По фронту **T2** выставленный адрес снимается с шины и выходы **nAD** переводятся в высокоимпедансное состояние, также выставляется активный низкий уровень на линии **nDIN**, далее микропроцессор переходит в состояние ожидания данных и сигнала подтверждения **nRPLY** от ведомого устройства. При этом запускается специальный таймер монитора шины, который осуществляет счет с частотой **CLC/8**. Если сигнал подтверждения **nRPLY** не будет получен в течение 56..64 тактов **CLC**, то цикл обмена прерывается и возникает программное исключение по вектору 000004<sub>8</sub> (или 160002<sub>8</sub> - зависит от состояния битов **PSW10** и **PSW11**). Разброс в величине таймаута связан с тем что фаза предделителя частоты **CLC/8** при запуске таймера не обнуляется - предделитель считает непрерывно.

На диаграмме в такте **T4** внешнее устройство выставляет данные, которые подлежат считыванию микропроцессором, и затем, по прошествии некоторого интервала формирует активный низкий уровень

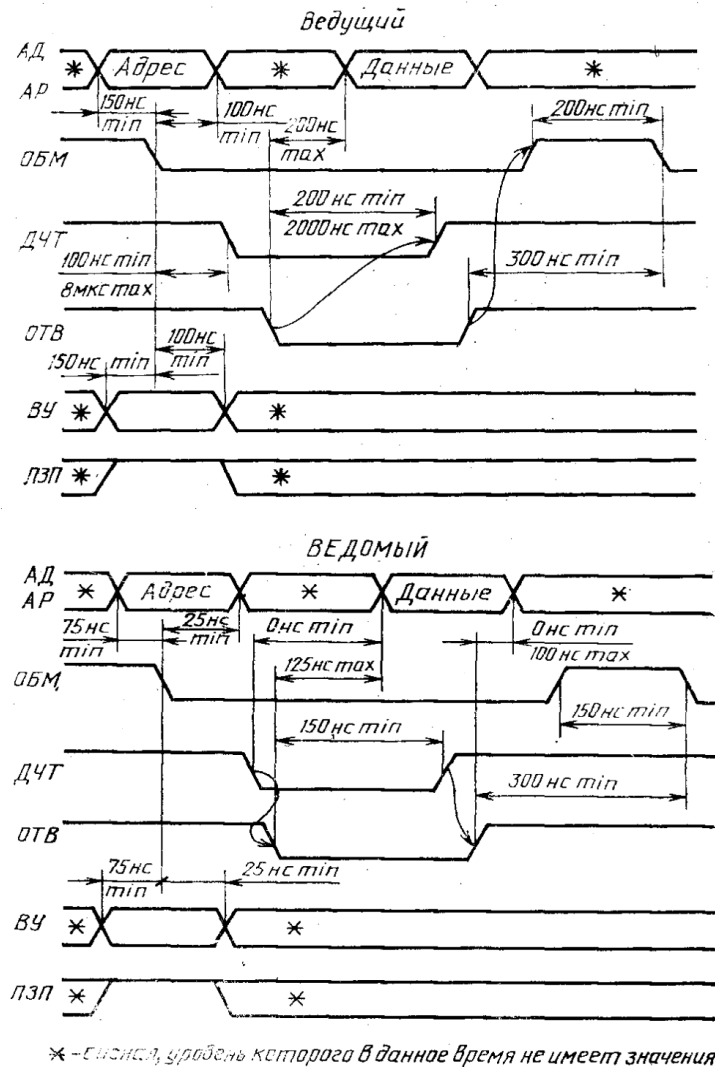
Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

сигнала подтверждения **nRPLY**. В общем же случае, согласно ГОСТ 26765.51-86, актуальные данные должны быть выставлены ведомым устройством не позднее чем 200 нс после среза активации сигнала **nRPLY**. На диаграмме момент выставления **nRPLY**, который определяется внешним устройством, намерено показан после фронта **T4**. Микропроцессор фиксирует сигнал на своем входе **nRPLY** по нарастающему фронту тактового сигнала, при обнаружении активного низкого **nRPLY** два фронта тактового сигнала подряд (на приведенной диаграмме это **T5** и **T6**, два цикла нужны так как данные от ведомого устройства могут запаздывать на интервал до 200 нс относительно активации **nRPLY**), цикл обмена полагается законченным, по фронту **T6** фиксируются читаемые данные на входах **nAD** и передаются внутренним схемам процессора, снимается активный сигнал **nDIN** и останавливается таймер монитора шины, затем происходит переход в ожидание снятия активного сигнала **nRPLY**. При этом детектирование неактивного высокого уровня **nRPLY** осуществляется по последовательным фронту и срезу тактового сигнала - сигнал на входе **nRPLY** должен быть неактивен сначала при нарастающем фронте **CLK**, а потом на срезе **CLK** следующего такта - таково условие срабатывания детектора. При обнаружении неактивного высокого уровня **nRPLY** запрещаются выходы управляющих сигналов **nDIN**, **nDOUT**, **nWTBT**. Далее, еще через один такт происходит снятие сигнала **nBSY** и выставление высокого уровня сигнала **nSYNC**. На диаграмме фронт сигнала **nRPLY** обнаруживается по срезу **T7**, а снятие сигналов **nSYNC** и **nBSY** происходит по срезу **T8**. Высокий неактивный уровень **nSYNC** удерживается в течение одного такта и по срезу **T9** выход переходит в высокоимпедансное состояние, микропроцессор перестает выполнять роль ведущего устройства, магистраль полагается незанятой. Следует отметить, что таймер монитора шины в состоянии ожидания деактивации **nRPLY** не работает, поэтому имеется теоретическая возможность зависания системы.

Микропроцессор K1801BM1 чтение данных всегда выполняет 16-битными словами. Если требуется прочитать только один из байтов, то ненужные разряды игнорируются. Также микропроцессор выставляет адрес "как есть" (например чтение по **@R0** выставляет на шину адреса непосредственно значение из регистра **R0**, **@#000001** - на шину будет выставлено значение 1 и так далее), и чтение слов по нечетному адресу может приводить к некорректно прочитанным данным, поскольку внешняя память и устройства обычно не поддерживают такой режим - в этом случае для корректного чтения внешним схемам необходимо прочитать из памяти два последовательных слова и собрать из них ожидаемую процессором комбинацию байтов. При чтении байта по четному адресу процессор ожидает данные на линиях **nAD0-nAD7**, при чтении байта по нечетному адресу данные ожидаются на линиях **nAD8-nAD15**.

Ниже в справочных целях приведена диаграмма выполнения транзакции "Чтение" из ГОСТ 26765.51-86 с указанием регламентируемых временных параметров.

Однокристальный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	



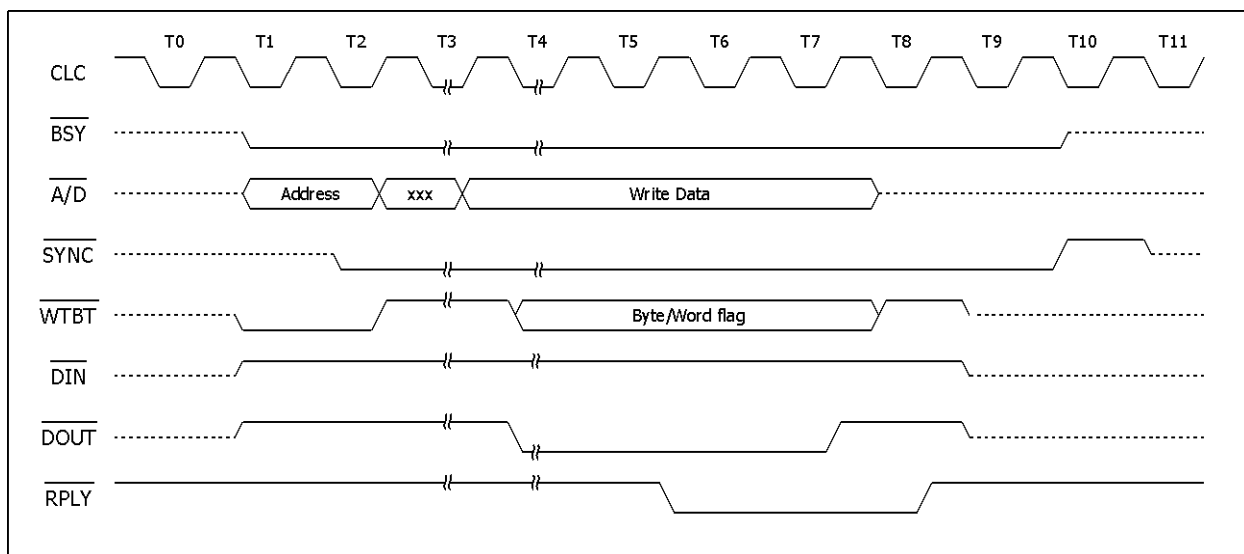


Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 5.2 Транзакция "Запись"

Ниже приведена диаграмма выполнения микропроцессором цикла записи данных по шине МПИ в качестве ведущего устройства.

Начало транзакции аналогично началу транзакции "Чтение". В такте **T0** на магистраль МПИ находится в исходном незнакомом состоянии, по срезу **T1** микропроцессор осуществляет захват магистрали, при этом разрешается управление выходами **nWTBT**, **nDIN**, **nDOUT**, **nIAKO**, **A/D**. Сигнал **nBSY** принимает активный низкий уровень, на шине адреса/данных устанавливается адрес записываемых данных. Выход сигнала **nSYNC** остается запрещенным (особенность K1801BM1). На выходе **nWTBT** установлен низкий уровень, что означает выполнение операции записи. По срезу **T2** на выходе **nSYNC** формируется активный низкий уровень. По фронту **T2** выставленный адрес снимается с шины, выходы **nAD** остаются разрешенными и на них транслируется содержимое внутреннего регистра данных микропроцессора. В данный момент там необязательно могут находиться актуальные записываемые в текущей транзакции данные - они могут быть еще не подготовлены внутренними схемами процессора. Время подготовки зависит от того какая программная инструкция в данный момент обрабатывается микропрограммой и может составлять от нуля до нескольких тактов. При готовности данных во внутреннем регистре они начинают транслироваться на внешние выходы по фронту тактового сигнала. Для краткости диаграммы полагаем, что данные выданы внутренними схемами в такте **T3**, практически же подготовка данных может занимать от нуля до нескольких тактов. По срезу следующего сигнала **T4** выставляется активный низкий уровень на линии **nDOUT** и значение признака байтовой операции **nWTBT**. Если происходит запись только одного из байт, то данный признак принимает значение активного низкого уровня. Запись, какого именно байта следует выполнить, определяется значением младшего бита адреса операции. Если происходит запись слова, то признак **nWTBT** принимает неактивное значение высокого уровня, выполнение операции записи слова по нечетному адресу приводит к некорректной записи данных. Далее микропроцессор переходит в состояние ожидания данных и сигнала подтверждения **nRPLY** от ведомого устройства. При этом запускается специальный таймер монитора шины, который осуществляет счет с частотой **CLC/8**. Если сигнал подтверждения **nRPLY** не будет получен в течение 56..64 тактов **CLC**, то цикл записи прерывается и возникает программное исключение по вектору 000004<sub>8</sub> (или 160002<sub>8</sub>).

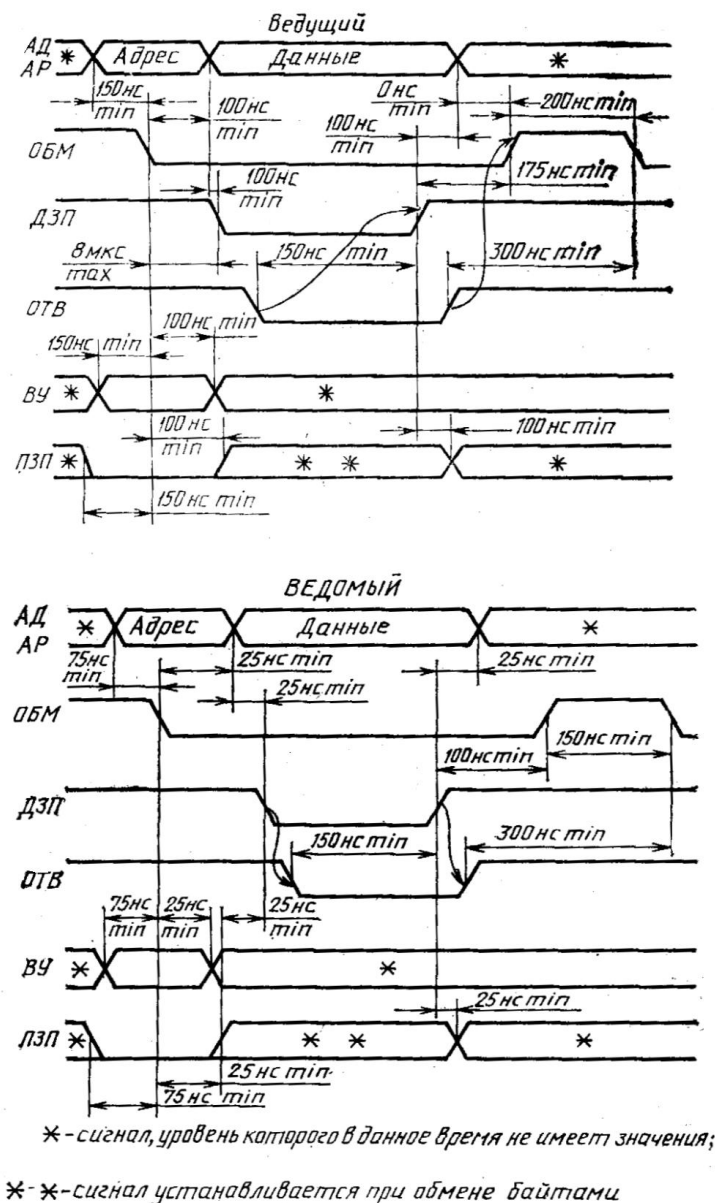


В такте **T5** внешнее устройство выполняет свою внутреннюю фиксацию записываемых данных с шины и затем, по прошествии некоторого интервала формирует активный низкий уровень сигнала подтверждения **nRPLY**. На диаграмме момент выставления **nRPLY**, который определяется внешним

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

устройством, намерено показан после фронта **T5**. Микропроцессор фиксирует сигнал на своем входе **nRPLY** по фронту тактового сигнала, при обнаружении активного низкого **nRPLY** два фронта тактового сигнала подряд (на приведенной диаграмме это **T6** и **T7**), цикл обмена полагается законченным. По фронту **T7** деактивируется сигнал **nDOUT** и останавливается таймер монитора шины, по срезу **T8** выходы шины данных A/D запрещаются, шина переходит в высокоимпедансное состояние. Завершение транзакции аналогично завершению транзакции "Чтение" - происходит переход в ожидание снятия активного сигнала **nRPLY**, при обнаружении деактивируются **nSYNC** и **nBSY**, магистраль переходит в неактивное состояние.

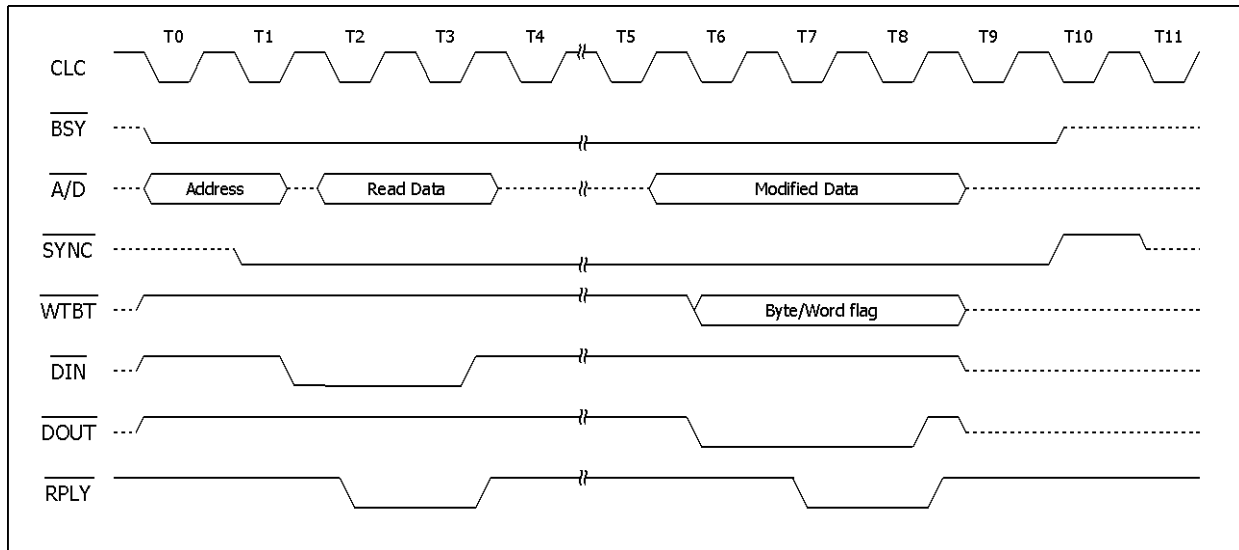
Ниже в справочных целях приведена диаграмма выполнения транзакции "Запись" из ГОСТ 26765.51-86 с указанием регламентируемых временных параметров



Однокристальный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

### 5.3 Транзакция "Чтение-Модификация-Запись"

Транзакция "Чтение-Модификация-Запись" имеет принципиальное значение для многопроцессорных систем, так как позволяет выполнять изменения переменных в памяти атомарно - за одну неразделяемую на части операцию. В этом ее принципиальное отличие от пары, состоящей из отдельной транзакции "Чтение" и последующей транзакцией "Запись" по тому же самому адресу. Ниже приведена диаграмма выполнения микропроцессором цикла чтение-модификация-запись по шине МПИ в качестве ведущего устройства.



Начало транзакции модификации данных полностью аналогично транзакции "Чтение". На диаграмме для большей компактности удален начальный такт с неактивной магистралью МПИ. В момент фиксации данных по фронту **T3** прочитанные данные передаются внутренним схемам микропроцессора, которые немедленно начинают над этими данными требуемую арифметическую или логическую операцию, а интерфейсный блок переходит в состояние ожидания деактивации сигнала **nRPLY**, при этом сигналы **nSYNC** и **nBSY** остаются активными. Эти действия выполняются блоками параллельно и независимо. Процессорному блоку может потребоваться несколько тактов для выполнения операции над данными, после ее завершения данные записываются в специальный внутренний регистр данных. После того как интерфейсный блок по срезу тактового сигнала обнаружит неактивный **nRPLY** и в регистре данных уже будут готовые модифицированные данные по фронту тактового сигнала (**T5** на диаграмме) будут разрешены выходы шины адреса/данных, и по срезу следующего тактового сигнала **T6** сформируется активный низкий уровень на выходе **nDOUT** и будет выставлен признак байтовой или словной операции. Завершение транзакции полностью аналогично завершению транзакции "Запись". Следует отметить что таймер монитора шины перезапускается отдельно для каждой из фаз транзакции и не работает в паузе между ними, когда ожидается готовность модифицированных данных от АЛУ и деактивация **nRPLY**.

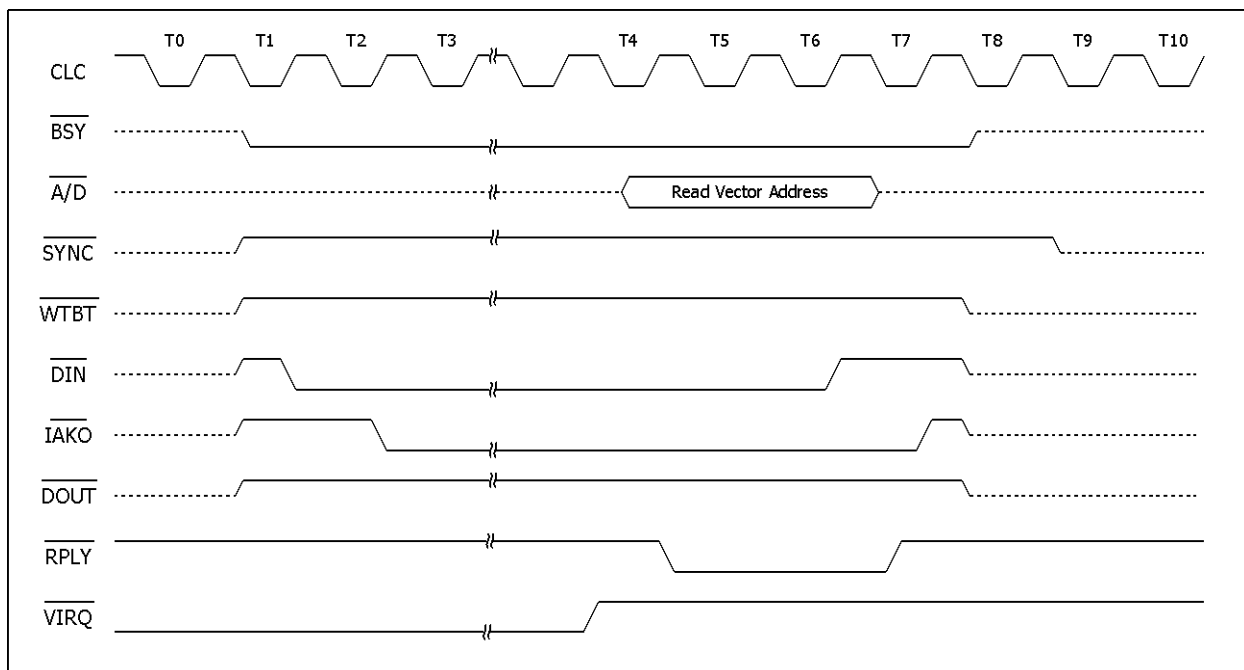


Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

#### 5.4 Транзакция "Подтверждение прерывания"

Ниже приведена диаграмма выполнения микропроцессором цикла чтения адреса вектора прерывания от внешнего устройства. Этот цикл выполняется процессором в рамках алгоритма обработки векторного прерывания после сохранения текущих PSW и PC в стеке. Пунктирными линиями показано состояние высокого импеданса.

Также указана привязка установки и снятия управляющих выходов к фронтам тактового сигнала, при этом не учитывается внутренняя задержка, которая может достигать десятков наносекунд и, при высоком значении тактовой частоты, может быть сравнимой с длительностью такта.



Допустим что в такте **T0** на магистрали МПИ нет ведущего устройства (однако обычно это не так, процессор сохранял PSW и PC согласно алгоритму входа в процедуру обработки прерывания), обмен не выполняется, полагаем это состояние исходным. По срезу **T1** микропроцессор осуществляет захват магистрали, при этом разрешается управление выходами **nWTBT**, **nDIN**, **nDOUT**, **nIAKO**. Сигнал **nBSY** принимает активный низкий уровень, сигнала **nSYNC** неактивен и остается таким на протяжении всей транзакции. На выходе **nWTBT** установлен высокий уровень, что означает выполнение операции чтения, выходы **nAD** находятся в высокоимпедансном состоянии, шина готова к приему данных от устройства. По фронту **T1** на выходе **nDIN** формируется активный низкий уровень. Через такт, по фронту **T2** на выходе **nIAKO** формируется низкий уровень и далее микропроцессор переходит в состояние ожидания данных (адреса вектора прерывания) и сигнала подтверждения **nRPLY** от ведомого устройства. При этом запускается специальный таймер монитора шины, который осуществляет счет с частотой **CLC/8**. Если сигнал подтверждения **nRPLY** не будет получен в течение 56..64 тактов **CLC**, то цикл обмена прерывается и возникает программное исключение по вектору 160012<sub>8</sub>. Разброс в величине таймаута связан с тем что фаза делителя частоты **CLC/8** при запуске таймера не обнуляется - делитель считает непрерывно.

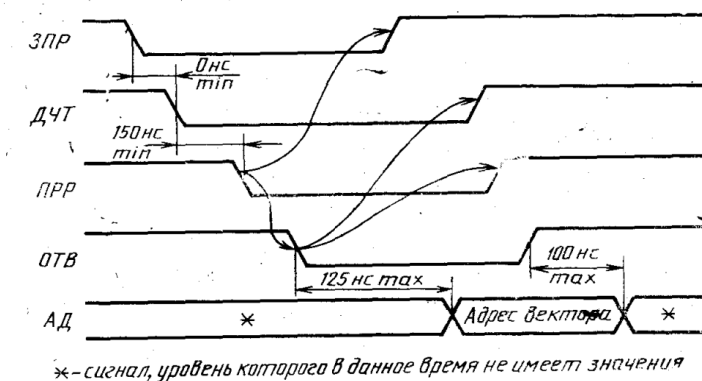
На диаграмме в такте **T4** внешнее устройство выставляет данные, которые подлежат считыванию микропроцессором, и затем, по прошествии некоторого интервала формирует активный низкий уровень сигнала подтверждения **nRPLY**. В общем же случае, согласно ГОСТ 26765.51-86, актуальные данные

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

должны быть выставлены ведомым устройством не позднее чем 200 нс после среза активации сигнала **nRPLY**. На диаграмме момент выставления **nRPLY**, который определяется внешним устройством, намерено показан после фронта **T4**. Микропроцессор фиксирует сигнал на своем входе **nRPLY** по фронту тактового сигнала, при обнаружении активного низкого **nRPLY** два фронта тактового сигнала подряд (на приведенной диаграмме это **T5** и **T6**, два цикла нужны так как данные от ведомого устройства могут запаздывать на интервал до 200 нс относительно активации **nRPLY**), цикл обмена полагается законченным, по фронту **T6** фиксируются читаемые данные и передаются внутренним схемам процессора, снимается активный сигнал **nDIN** и останавливается таймер монитора шины, затем в такте **T7** деактивируется сигнал **nIAKO** и происходит переход в ожидание снятия активного сигнала **nRPLY**. При этом по срезу тактового сигнала осуществляется детектирование неактивного высокого уровня **nRPLY** и при обнаружении такового запрещаются выходы управляющих сигналов **nDIN**, **nDOUT**, **nWTBT**, **nIAKO**.

Ведомое устройство которое выставило вектор на шину может снять активный запрос **nVIRQ** если у него больше нет других запросов на прерывание.

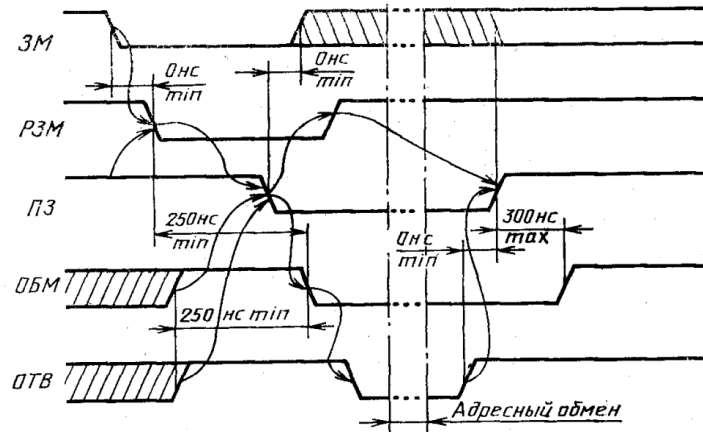
Ниже в справочных целях приведена диаграмма выполнения транзакции "Чтение вектора прерывания" из ГОСТ 26765.51-86 с указанием регламентируемых временных параметров.



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 5.5 Процедура арбитража доступа к магистрали

Ниже в справочных целях приведена диаграмма выполнения транзакции "Захват магистрали" из ГОСТ 26765.51-86 с указанием регламентируемых временных параметров.



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 5.6 Поддержка многопроцессорной конфигурации

Процессор 1801BM1 поддерживает работу в многопроцессорной конфигурации, которая позволяет строить системы, содержащие до четырех процессоров 1801BM1, непосредственно подключенных к общей шине. Каждому процессору в такой системе присваивается аппаратный номер от 0 до 3. Аппаратный номер процессора определяется комбинацией сигналов на входах **nPA1** и **nPA0**:

**nPA1=1, nPA0=1** - процессор номер 0, высший приоритет доступа к шине

**nPA1=1, nPA0=0** - процессор номер 1

**nPA1=0, nPA0=1** - процессор номер 2

**nPA1=0, nPA0=0** - процессор номер 3

Если процессор единственный в системе (однопроцессорный вариант), то на данные входы должен быть подан высокий уровень. Процессор номер 0 называется ведущим, процессоры 1-3 называются ведомыми. Ведущий процессор всегда должен быть на вершине цепочки **nDMGI/nDMGO** и имеет наивысший приоритет доступа к шине, порядок ведомых процессоров в цепочке не важен.

От аппаратного номера процессора зависит следующий функционал:

- базовый адрес блока регистров внутреннего периферийного блока, периферийный блок будет отвечать на обращения по шине только по соответствующим адресам
- адрес первого обращения к регистру начального пуска после аппаратного сброса
- биты 9 и 8 слова состояния процессора **PSW**
- биты 4 и 3 регистра управления режимом  $1777x0_8$
- ведущий процессор никогда не активирует **nDMR**
- ведомые процессоры игнорируют вход **nVIRQ**
- различная логика арбитража **nDMGI/nDMGO** для ведомых и ведущего процессоров

Адрес регистра используемого для переключения пользовательского и пультового режимов постоянен ( $177716_8$ ) и не зависит от номера процессора, также постоянен адрес ячеек для сохранения/извлечения **PC** и **PSW** пультового режима ( $177674_8$ ). Физическое разделение этих общих ресурсов в многопроцессорной системе может быть возложено на внешние схемы.

Для программной синхронизации доступа к общим ресурсам предполагается использование инструкций выполняющих обращение к шине при помощи транзакции чтение-модификация-запись. Процессор 1801BM1 выполняет такую транзакцию на шине МПИ атомарно, с монопольным захватом шины на все время выполнения транзакции. Например, процедуры захвата и освобождения синхронизирующего объекта (spinlock) могут быть основаны на инструкциях **DECB** или **ASRB**:

```
lock:
    cmpb    spinlock, #1
    bne     lock
    decb    spinlock
    bne     lock
    rts     PC

unlock:
    mov     #1, spinlock
    rts     PC

lock:
    asrb    spinlock
    bcc     lock
    rts     PC

unlock:
    mov     #1, spinlock
    rts     PC
```



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 6. Периферийный блок

Внутри микросхемы K1801 также имеется периферийный блок, который подключен к внешним выводам магистрали МПИ и может играть роль встроенного ведомого устройства магистрали. Это означает что регистры периферийного блока могут читаться и записываться по МПИ как внутренним процессорным блоком микросхемы, так и внешним по отношению к микросхеме агентом шины, например, другим процессором.

Базовый адрес периферийного блока зависит от физического номера процессора в системе (задается значениями на входах **nPA0**, **nPA1**):

Входы	Номер процессора	Диапазон адреса
<b>nPA1=1, nPA0=1</b>	процессор номер 0	177700 <sub>8</sub> -177716 <sub>8</sub>
<b>nPA1=1, nPA0=0</b>	процессор номер 1	177720 <sub>8</sub> -177736 <sub>8</sub>
<b>nPA1=0, nPA0=1</b>	процессор номер 2	177740 <sub>8</sub> -177756 <sub>8</sub>
<b>nPA1=0, nPA0=0</b>	процессор номер 3	177760 <sub>8</sub> -177776 <sub>8</sub>

При обращении к диапазону адресов базового блока процессор всегда генерирует ответный сигнал **nRPLY**, не требуется его формирование дополнительными внешними схемами. При обращении по адресу база внутреннего периферийного блока + 16<sub>8</sub> дополнительно генерируется активный внешний сигнал **nSEL1** низкого уровня, независимо от того какой агент, внутренний или внешний обращается к данному адресу. При обращении по адресу база внутреннего периферийного блока + 14<sub>8</sub> дополнительно генерируется активный внешний сигнал **nSEL2**. Сигналы **nSEL1** и **nSEL2** фиксируются по ниспадающему фронту **nSYNC** и остаются неизменными на протяжении всего цикла обмена шины до нарастающего фронта **nSYNC**.

При обращении к периферийному блоку по смещениям 000<sub>8</sub>-012<sub>8</sub> данные выдаются и принимаются внутренними схемами микропроцессора - регистрами управления и ВЕ-таймера. При обращении по смещениям 014<sub>8</sub>-016<sub>8</sub> данные должны быть выданы или приняты внешними схемами. Следующая таблица раскрывает особенности обращений к внутреннему периферийному блоку микросхемы:

Операция	Агент шины	Адрес обращения	Линии <b>nAD</b>	<b>nRPLY</b>
чтение	внутренний	вне диапазона блока	вход	внешний
		база + 000 <sub>8</sub> -012 <sub>8</sub>	выход	внутри
		база + 014 <sub>8</sub> -016 <sub>8</sub>	вход	внутри
	внешний	вне диапазона блока	вход	внешний
		база + 000 <sub>8</sub> -012 <sub>8</sub>	выход	внутри
		база + 014 <sub>8</sub> -016 <sub>8</sub>	вход	внутри
запись	внутренний	вне диапазона блока	вход	внешний
		база + 000 <sub>8</sub> -012 <sub>8</sub>	выход	внутри
		база + 014 <sub>8</sub> -016 <sub>8</sub>	вход	внутри
	внешний	вне диапазона блока	вход	внешний
		база + 000 <sub>8</sub> -012 <sub>8</sub>	вход	внутри
		база + 014 <sub>8</sub> -016 <sub>8</sub>	вход	внутри

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Под внутренним агентом понимается собственный процессорный блок микросхемы, под внешним - внешний процессор или другое устройство, осуществившее захват шины и являющееся в момент обращения ведущим. Таблица также показывает состояние линий **nAD**, при состоянии выход на шину транслируется содержимое соответствующего регистра, из которого осуществляется чтение, или данные, записываемые в регистр внутренним процессорным блоком. Последний столбец отображает формирование сигнала **nRPLY**, при обращении к периферийному блоку сигнал **nRPLY** формируется всегда.

Регистры периферийного блока можно условно разделить на три группы:

- регистры управления и состояния - смещения 000<sub>8</sub>-004<sub>8</sub>
- регистры ВЕ-таймера - смещения 006<sub>8</sub>-012<sub>8</sub>
- внешние регистры **nSEL2** и **nSEL1** - смещения 014<sub>8</sub>-016<sub>8</sub>

Данные регистры описаны в следующих разделах.

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 6.1 Регистр управления режимом

<b>Name:</b> CPU_MODE															
<b>Offset:</b> 177700 <sub>8</sub> + <номер процессора> * 20 <sub>8</sub>								<b>Reset Value:</b> 177740 <sub>8</sub> (зависит от номера)							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Readonly (Write ignored)												R & W			
1	1	1	1	1	1	1	1	1	1	1	PA1	PA0	WT	ST	HLT

Регистр режима, начальное значение 177740<sub>8</sub> (зависит от аппаратного номера процессора)

Биты 0,1,2 регистра режима используются для синхронизации пуска подчиненных процессоров главным процессором в многопроцессорной системе, т.к. блок регистров каждого процессора доступен на шине любому другому процессору вне зависимости от состояния процессора-владельца

**HLT** - останов конвейера обработки микрокоманд. Процессор останавливается полностью, выход из этого состояния возможен только при сбросе этого бита, выполняемого внешним агентом на шине МПИ путем прямой записи в регистр или по аппаратному сбросу микропроцессора. При выполнении микрокода это бит также может устанавливаться записью 0 в 25 разряд специального управляющего вектора, но рабочая версия микропрограммы эту возможность никогда не использует.

**ST** - остановить конвейер микроинструкций после загрузки регистра инструкций. Если этот бит имеет единичное значение, то при загрузке слова кода очередной команды в регистр инструкций, происходит установка бита **HLT** и конвейер останавливается, таким образом, возможно пошаговое исполнение программы под управлением внешнего агента по шине МПИ.

**WT** - при установке этого бита происходит переход в режим ожидания прерывания, бит автоматически устанавливается инструкцией WAIT. При наступлении разрешенного прерывания этот бит сбрасывается и происходит переход из режима ожидания прерывания в рабочий режим. Физически этот бит совпадает с флагом **WCPU** в блоке прерываний. Может управляться микропрограммой через запись в специальный управляющий вектор.

**PA1, PA0** - аппаратный номер процессора в многопроцессорной системе, для ведущего процессора в данном поле считывается значение 0, для ведомых - 1,2 или 3.

Однокристальный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 6.2 Регистр вектора прерывания рестарта

Name: CPU_IVEC															
Offset: $177702_8 + \text{<номер процессора>} * 20_8$								Reset Value: N/A							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write only (Read as $177777_8$ )															
Unknown															

Регистр адреса прерывания, начальное значение  $177777_8$ , только для записи

Данный регистр представляет собой рудиментарный остаток от K1801BE1. Предполагалось что при записи в данный регистр записываемое значение будет сохраняться и при этом активироваться запрос на прерывание. Пока данный запрос на прерывание не будет обработан, регистр вектора рестарта делается более недоступным по шине - при обращении к нему не генерируется **nRPLY**, и также запись нового значения не осуществляется. Таким образом внешний агент может узнать что процессор-владелец регистра еще не обработал прерывание. Обработка возникшего запроса предполагалась по вектору, значение которого было записано в данный регистр рестарта. Однако ни K1801BM1A ни K1801BM1Г не реализует полный функционал запроса на прерывание от регистра вектора рестарта - данный запрос просто не подключен к блоку обработки прерываний. При записи произвольных данных в регистр он "пропадает" из адресного пространства, так как внутренний запрос на прерывание установлен и блокирует генерацию **nRPLY**. Но, как выяснилось, процессоры K1801BM1 ошибочно сбрасывают этот запрос на прерывание при обработке инструкции **EMT**, таким образом регистр вектора рестарта становится снова доступен на шине.

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

### 6.3 Регистр флагов ошибок

Name: CPU_ERROR															
Offset: 177704 <sub>8</sub> + <номер процессора> * 20 <sub>8</sub>								Reset Value: 177440 <sub>8</sub>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Readonly (Write ignored)								Read & Write							
1	1	1	1	1	1	1	1	ER7	OAT	1	QTO	ER3	ER2	UOP	DBL

Регистр ошибки, начальное значение после сброса 177440<sub>8</sub> - все используемые флаги ошибок очищены. Единичное значение флага означает что имела место соответствующая ошибочная ситуация.

**DBL** - double error - двойная ошибка (повторная установка битов **UOP**, **QTO**, **OAT**, **ER7**)

**UOP** - unknown opcode - недопустимая команда, вызывает сброс микропрограммного автомата

**ER2** – резервный бит, не вызывает исключений, никогда не устанавливается микрокодом

**ER3** – резервный бит, не вызывает исключений

**QTO** - qbus timeout - тайм-аут обращения к внешнему устройству

**OAT** - odd address trap - словное обращение по нечетному адресу, поскольку детектор обращения по нечетному адресу в 1801BM1 аппаратно заблокирован, то данный бит никогда не устанавливается.

**ER7** - резервный бит, не вызывает исключений, но может вызвать двойную ошибку

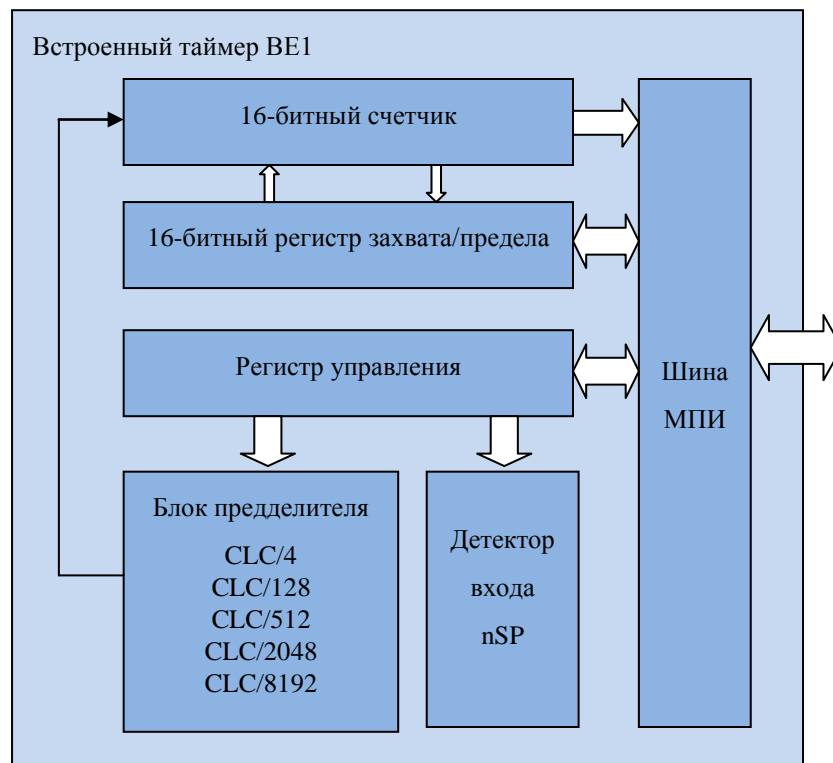
Все биты ошибок безусловно сбрасываются при загрузке нового кода команды во внутренний регистр инструкций, таким образом, регистр ошибок отображает ошибочные ситуации возникшие при выполнении текущей команды. Поэтому значение некоторых битов ошибок не может быть прочитано программным способом самим процессором - инструкция успешного чтения данных из самого регистра выполняется без ошибок **DBL**, **UOP**, **QTO**, **OAT**, эти биты будут сброшены. Но регистр ошибок может быть прочитан внешним агентом на шине МПИ (например, другим процессором) и полученная информация может быть использована в отладочных и диагностических целях.

Биты **ER2**, **ER3**, **ER7** могут устанавливаться микропрограммой через запись в специальный управляющий вектор, но на практике рабочей версией микропрограммы эта возможность не используется, вероятно эти флаги могли использоваться при отладке микропрограммы.

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

#### 6.4 Регистры встроенного таймера K1801BM1

Встроенный таймер K1801BM1 является унаследован от однокристалльного микроконтроллера K1801BE1. Таймер является недокументированным, и некоторый процент микросхем K1801BM1A/Б/В имеют неработоспособный блок встроенного таймера. Микросхема K1801BM1Г имеет гарантировано работоспособный блок таймера, а также дополнительно поддерживает прерывание от этого блока.



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Таймер имеет в своем составе следующие блоки:

- 16-битный декрементируемый счетчик
- 16-битный регистр значения перезагрузки и/или захвата текущего значения счетчика
- регистр управления таймером
- регистр управления
- предделитель входной процессорной частоты **CLC** на **4**
- дополнительный делитель частоты **CLC** на  $128_{10}$  (формируется делением **CLC/4** на **32**)
- детектор ниспадающего фронта на входе **nSP**

Тактирование счетчика таймера может осуществляться одной из следующего набора частот:

- внешняя тактовая частота, поступающая на внешний вход микросхемы **nSP**
- частота с предделителя **CLC/128**
- частота с предделителя **CLC/128** дополнительно деленная на **4**
- частота с предделителя **CLC/128** дополнительно деленная на **16**
- частота с предделителя **CLC/128** дополнительно деленная на **64**

Также в процессоре K1801BM1Г таймер может генерировать прерывания по вектору 270<sub>8</sub>, в процессоре K1801BM1А данная функция отключена, прерывания о таймера не генерируются

Все внутренние управляющие схемы таймера тактируются частотой **CLC/4**. Предделитель и делители **CLC** программно недоступны, работают постоянно и не сбрасываются при обращении к регистрам таймера или при сбросе всего процессора по входу **nDCLO**.

Для внешней частоты **nSP** счетный такт формируется по ниспадающему фронту сигнала, синхронизированного с частотой **CLC/4**, поэтому значение входной частоты не может быть больше чем **CLC/8**, а длительность любого полупериода должна превышать длительность периода **CLC/4**.

Таймер может быть настроен на следующие режимы:

- декрементный счет с остановом по достижению счетчика нулевого значения
- декрементный счет с перезагрузкой счетчика значением из регистра перезагрузки при достижении счетчика нулевого значения с последующим продолжением счета
- декрементный счет с захватом текущего значения счетчика в регистре перезагрузки при обнаружении ниспадающего фронта на входе **nSP**. В этом режиме счетчик не останавливается при достижении нуля, а переходит на значение  $177777_8$ . Обратное копирование из регистра перезагрузки в регистр счетчика при достижении счетчиком нуля в данном режиме не выполняется.

Однокристальный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 6.5 Регистр значения перезагрузки и захвата

<b>Name:</b> TVE_LIMIT															
<b>Offset:</b> 177706 <sub>8</sub> + <номер процессора> * 20 <sub>8</sub>								<b>Reset Value:</b> N/A							
<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
Read & Write															
Limit															

**Limit** – 16-битное начальное значение счета

Может быть настроен режим, при котором в данный регистр копируется содержимое регистра счета по ниспадающему фронту на внешнем входе **nSP**.

Вне зависимости от режима работы, при записи со стороны процессора в регистр управления таймера **TVE\_CSR** содержимое регистра **TVE\_LIMIT** копируется в регистр счетчика **TVE\_COUNT**.

Начальное значение регистра после включения питания не определено, зависит от того в какое псевдослучайное состояние установятся триггеры регистра. Сброс по входам **nINIT** и **nDCLO** никак не изменяет состояние регистра **TVE\_LIMIT**.



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 6.6 Регистр счетчика таймера

Name: TVE_COUNT															
Offset: 177710 <sub>8</sub> + <номер процессора> * 20 <sub>8</sub>								Reset Value: N/A							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Readonly (Write Ignored)															
Count															

**Count** – 16-битное текущее значение счетчика таймера

Декрементируется с частотой **nSP**, или **CLC/128** опционально деленной на **4**, **16** или **64**. Если настроен режим перезагрузки счетчика значением из регистра **TVE\_LIMIT**, то перезагрузка осуществляется в течение 4 тактов **CLC** (не **CLC/128**) после перехода счетчика из значения 1 в значение 0, таким образом, счетчик последовательно принимает следующие значения: N, N-1, N-2, ..., 3, 2, 1, (кратковременный 0), N, N-1... Фаза предделителя и делителей, а также фаза детектора ниспадающего фронта **nSP** при этом сохраняется неизменной и равномерность периода счета не нарушается. Для простоты программной модели можно считать что значение **0** пропускается, хотя при некотором стечении обстоятельств и может быть прочитано из регистра.

Вне зависимости от режима работы таймера, при каждой записи в регистр управления **TVE\_CSR** происходит переписывание значения регистра **TVE\_LIMIT** в регистр **TVE\_COUNT**.

При условии, что не установлен бит **CAP** в регистре **TVE\_CSR** происходит загрузка счетчика значением из регистра **TVE\_LIMIT** при достижении счетчиком нулевого значения. Тут имеется особенность - загрузка происходит только в том случае, если счетчик перед этим был ненулевым (имел единичное значение и был декрементирован), если же в **TVE\_LIMIT** записано нулевое значение, то оно загружается один раз, далее условие ненулевого значения счетчика в предыдущем такте не выполняется и новой загрузки не происходит. В следующем такте счета, счетчик принимает значение **177777<sub>8</sub>** и продолжает дальнейший счет.

Начальное значение регистра после включения питания не определено, зависит от того в какое псевдослучайное состояние установятся триггеры регистра. Сброс по входам **nINIT** и **nDCLO** никак не изменяет состояние регистра **TVE\_COUNT**.

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 6.7 Регистр управления таймера

<b>Name:</b> TVE_CSR															
<b>Offset:</b> 177712 <sub>8</sub> + <номер процессора> * 20 <sub>8</sub>								<b>Reset Value:</b> 177400 <sub>8</sub>							
<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
Readonly								Read & Write							
1	1	1	1	1	1	1	1	FL	D4	D16	RUN	OS	MON	CAP	SP

**FL** – Аппаратно устанавливается схемой таймера в единицу в случае наступления события таймера. Событием таймера может быть момент перехода счетчика через нуль или захват текущего значения счетчика по ниспадающему фронту на входе **nSP**. Для установки бита **FL** должен быть разрешен мониторинг события таймера - бит **MON** должен быть установлен программой в единичное значение. На самом деле, при наступлении события таймера (переход счетчика через нуль или захват значения счетчика) генерируется запрос на прерывание, при запуске обработки прерывания данный запрос сбрасывается. Бит **FL** устанавливается только в том случае если возник новый запрос на прерывание а старый еще не был обработан. Поскольку функция обслуживания прерывания от таймера в K1801BM1A заблокирована, то бит **FL** будет реально установлен только после второго срабатывания таймера. После включения питания или сброса **nDCLO** при самом первом событии таймера данный бит не устанавливается. При втором и последующих событиях этот бит работает правильно, поскольку внутренний флаг запроса на прерывание уже установлен. Для процессора 1801BM1Г возможна обработка прерываний от таймера, и бит **FL** будет выполнять роль флага переполнения - будет устанавливаться при возникновении события если прерывание от предыдущего события еще не обработано

**D4** – разрешить делитель на 4, делители 1/4 и 1/16 включаются последовательно

**D16** – разрешить делитель на 16, делители 1/4 и 1/16 включаются последовательно

**RUN** – запись единицы разрешает работу счетчика, значение из **TVE\_LIMIT** копируется в **TVE\_COUNT** и далее значение счетчика будет уменьшаться на единицу с заданной частотой. Запись нуля останавливает счетчик (при этом значение из **TVE\_LIMIT** также копируется в **TVE\_COUNT** как и при любой записи в регистр управления). Если установлен бит **OS** то при достижении нулевого значения счетчика данный бит будет сброшен аппаратно схемой таймера. Бит **RUN** сбрасывается только событием перехода через нуль при сброшенном бите **CAP**. По событию захвата при ниспадающем фронте на входе **nSP** и установленном бите **CAP** счетчик не останавливается и всегда продолжает счет.

**OS** – использовать однократный режим, при достижении счетчиком нуля, бит **RUN** будет сброшен, таймер перезагружен и остановлен. Для использования этого режима также необходимо сбросить бит **CAP**, т.е. если оба бита **OS** и **CAP** установлены, то таймер не остановится при переходе счетчика через нуль, а будет продолжать отсчет, так как установленный бит выбирает генерацию события таймера по ниспадающему фронту на входе **nSP**.

**MON** – разрешить мониторинг события таймера - "переход счетчика через нуль" или "захват значения счетчика", устанавливать бит **FL** при возникновении события таймера. Также для 1801BM1Г данный бит разрешает запрос прерывания от таймера.

**CAP** - режим захвата, по ниспадающему фронту на входе **nSP** осуществить копирование регистра счетчика **TVE\_COUNT** в регистр предела **TVE\_LIMIT**, при этом генерируется событие таймера. Бит **RUN** также должен быть установлен, иначе событие не генерируется. При нулевом значении бита **CAP** таймер генерирует событие при переходе счетчика через значение нуль и осуществляет загрузку регистра счетчика из регистра предела.

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

**SP** - выбор режим тактирования счетчика таймера по ниспадающему фронту на входе **nSP**, делители на 4 и на 16 в данном случае никак не влияют на результирующую частоту тактирования таймера.

**Таблица частоты тактирования счетчика таймера**

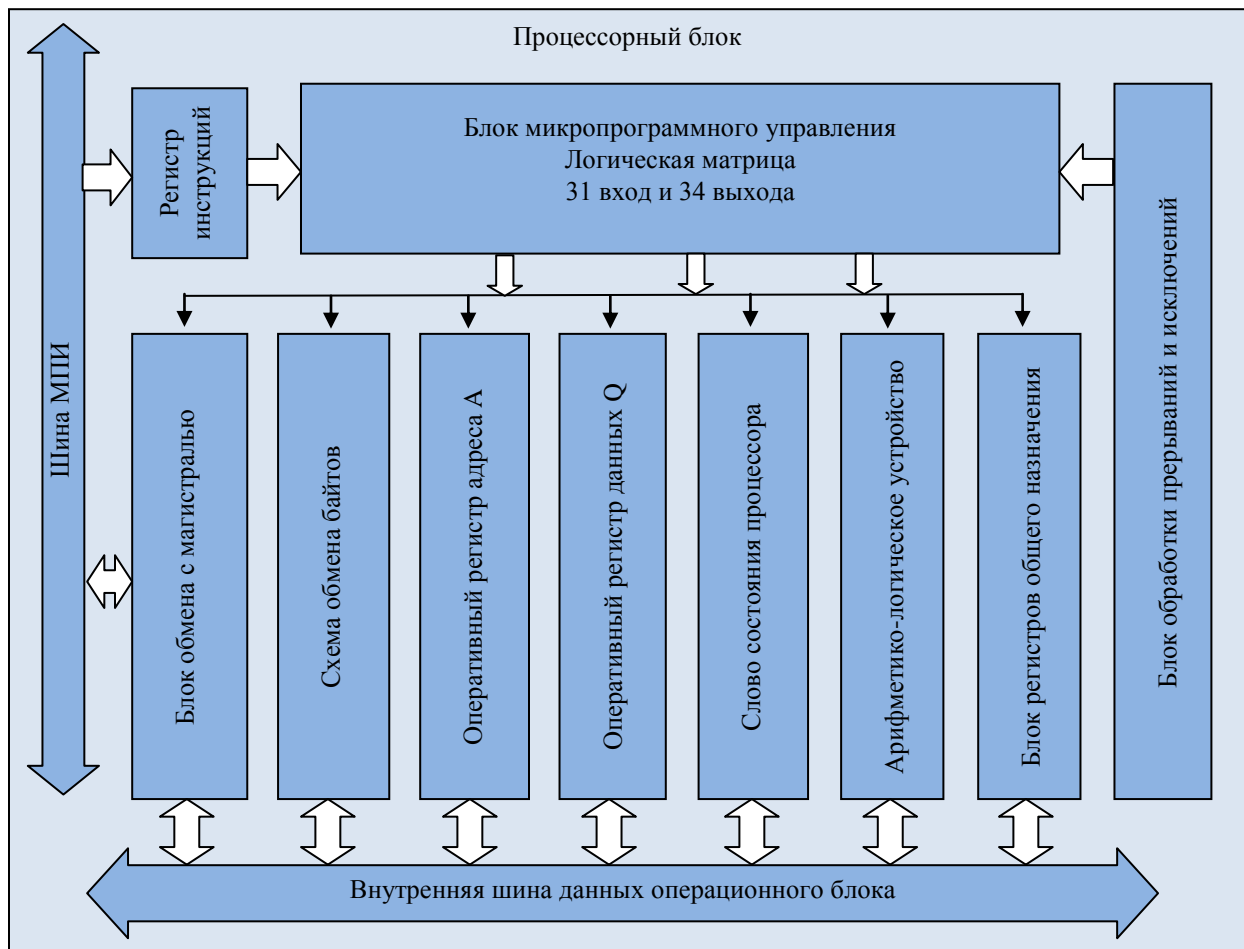
D4	D16	SP	Частота
x	x	1	вход <b>nSP</b> ( $CLC/8_{10}$ максимальная)
0	0	0	$CLC/128_{10}$
0	1	0	$CLC/2048_{10}$
1	0	0	$CLC/512_{10}$
1	1	0	$CLC/8192_{10}$

**Режимы работы таймера**

RUN	OS	CAP	Режим
0	x	x	Остановлен
1	0	0	Непрерывный счет с перезагрузкой счетчика
1	1	0	Однократный счет с предзагрузкой счетчика
1	x	1	Непрерывный счет без перезагрузки счетчика, захват значения счетчика по входу <b>nSP</b>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 7. Процессорный блок



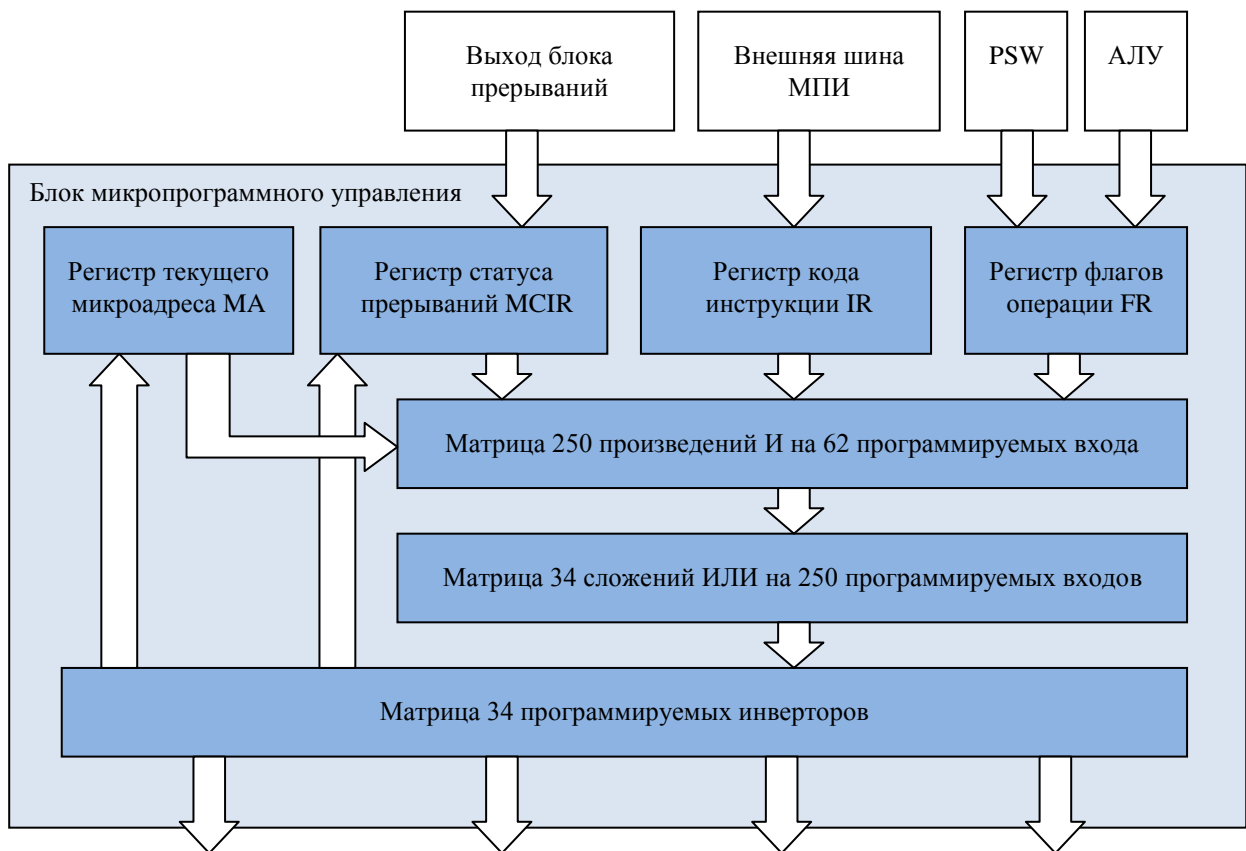
В составе процессорного блока можно выделить следующие основные компоненты:

- блок микропрограммного управления, осуществляет декодирование инструкций и формирование необходимых последовательностей сигналов для их выполнения в других обрабатывающих блоках процессора, использует отдельный внутренний регистр инструкций, в котором хранится код исполняемой инструкции
- блок обработки прерываний и исключений, получает на входы все имеющиеся сигналы прерываний и исключений, сигналы их маскирования или запрета, производит их упорядочивание по приоритету и передает информацию о текущем наиболее приоритетном событии блоку микропрограммного управления
- операционный блок, выполняет собственно обработку и хранение данных, содержит в своем составе отдельную внутреннюю шину данных, блок взаимодействия с шиной МПИ, схему обмена байтов, оперативный регистр данных **Q**, оперативный регистр адреса транзакции МПИ **A**, слово состояния процессора **PSW**, арифметико-логическое устройство и блок регистров общего назначения

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 7.1 Блок микропрограммного управления

Ниже приведена диаграмма блока микропрограммного управления и источники входных сигналов данного блока. Выходы блока управляют работой практически всех остальных блоков процессора.



В блоке микропрограммного управления можно выделить следующие компоненты:

- регистр микроадреса **МА**, этот регистр содержит 7-битное значение адреса текущей микроинструкции. При аппаратном сбросе процессора регистр микроадреса безусловно принимает значение **00**, обеспечивая начальный пуск. При возникновении ситуации микросброса (внутренний сброс микропрограммного автомата, возникает по ряду причин - недопустимый код инструкции, тайм-аут ввода-вывода и прочее) регистр принимает значение **01**, обеспечивая прекращение выполнения текущей машинной инструкции и принудительный безусловный переход к последовательности микроинструкций обработки исключений. Во всех остальных случаях регистр микроадреса принимает значение адреса следующей микроинструкции в последовательности, формируемое на части выходов логической матрицы
- регистр статуса прерывания **MCIR**, этот регистр содержит 3-битное значение, обычно формируемое блоком обработки прерываний и исключений. Также значение может формироваться с выходов логической матрицы, это используется при обработке инструкций программных исключений типа HALT, IOT и тому подобных

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

- регистр инструкций **IR**, 16-битный регистр, содержащий слово текущей машинной инструкции, записывается исключительно результатом транзакций чтения внешней шины. На некоторых этапах работы микропрограммного автомата значение этого регистра игнорируется, например, при начальном пуске или обработке неопределенного кода инструкции
- регистр флагов **FR**, внутренний 5-битный регистр, старший разряд всегда принимает значение четвертого разряда слова состояния процессора (флаг **T**), остальные разряды могут также загружаться из младших битов **PSW**, либо получать значения арифметических флагов напрямую от АЛУ. Это сделано для возможной микропрограммной реализации различных сложных инструкций типа **MUL**, без модификации программно-видимых арифметических флагов
- универсальная программируемая логическая матрица на 31 вход и 34 выхода. Матрица реализует 34 универсальных однобитовых функции от 31 входных аргументов. Время прохождения сигналов через матрицу составляет два такта частоты **CLC**. В моменты времени когда работа матрицы не требуется (например, идет ожидание завершения операции чтения на шине МПИ) с транзисторов матрицы снимается питание, что приводит к снижению потребления структурами n-МОП. Матрица содержит три ступени – логические умножения, логические сложения и выходные инверторы. Ступень логических умножений содержит 250 функций логических произведений от 31 входного аргумента и 31 инверсии входных аргументов. Под программированием матрицы понимается что в функции произведений могут быть включены входные аргументы и их инверсии в произвольном порядке, или не подключены вовсе. Подключение аргументов производится масочным способом при производстве микросхемы и не может быть изменено в процессе дальнейшей эксплуатации. Вторая ступень содержит 34 функции логического сложения каждая от 250 аргументов, аргументами являются выходы функций логического умножения первой стадии. Функции логического сложения второй стадии также являются программируемыми – масочным способом задаются подключаемые аргументы. Выходы функций второй ступени поступают на программируемые 34 инвертора, инверсия которых также задается масочным способом при изготовлении микросхемы. Такая структура матрицы позволяет реализовать независимые варианты 34 комбинационных функций. Микропроцессоры 1801BM1A и 1801BM1Г имеют различные варианты программирования матрицы, соответственно в них реализованы различные варианты поведения микропрограммного автомата

Функционирование блока микропрограммного управления основано на переборе микроадресов. Как таковых микроинструкций нет, есть набор микроадресов, загружаемых в регистр **MA**. Значение этого регистра и ряд других аргументов подается на вход матрицы, матрица формирует среди прочих значений поле следующего микроадреса **NA** – это значение, которое будет загружено в конце цикла в регистр **MA**.

Цикл работы микропрограммного автомата занимает минимум 2 такта входной частоты **CLC**, но может быть удлинен на произвольное количество тактов по ряду причин:

- ожидание готовности результата АЛУ
- ожидание завершения чтения данных на шине МПИ, когда данные нужны в качестве аргумента для операции АЛУ, данные будут занесены в регистр **Q** и использованы в операции
- ожидания завершения чтения кода инструкции на шине МПИ и записи его в регистр инструкций **IR**
- ожидание завершения опроса блока прерываний

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

- ожидание освобождения регистра адреса **A** для инициирования новой транзакции на шине МПИ в конвейере

В каждом цикле работы микропрограммного автомата АЛУ принимает два входных аргумента с шин **X** и **Y** и выполняет заданную арифметическую или логическую операцию, результат которой выдается на шину **X** и опционально может быть записан или в регистре общего назначения, или в регистр слова состояния **PSW**, или в регистры адреса **A** и данных **Q**. В регистры адреса или данных опционально может осуществляться запись аргумента с шины **X** до выполнения операции АЛУ, например так запускается чтение слова кода очередной инструкции по значению счетчика **PC**, при этом в данном цикле АЛУ выполняет операцию сложения **PC** (блок РОН выдает на шину **X**) + 2 (генератор констант выдает на шину **Y**), но в регистр адреса будет записано значение еще до модификации, это позволяет уменьшить количество циклов микропрограммного автомата, нужных для обработки процессорных инструкций. Также в каждом цикле может быть опционально запущен опрос блока прерываний, который завершается записью регистров **MCIR** и **VSEL**. Эти регистры могут быть записаны также непосредственно в ходе обработки некоторых инструкций (EMT, TRAP, и так далее) и использованы для последующего вызова обработки исключений.

Характерные последовательности перебора микроадресов при обработке прерываний и исключений, а также исполнении процессорных инструкций приведены в разделе описания микропрограммы.

#### Назначение выходов блока микропрограммного управления

00	NA6	Адрес следующей микроинструкции, значение формируемое данными выходами будет загружено в регистр текущего микроадреса <b>MA</b> после выполнения текущей микроинструкции. При аппаратном сбросе процессора или внутреннем сбросе микропрограммного автомата данное поле игнорируется
05	~NA5	
09	~NA4	
15	~NA3	
19	NA2	
24	NA1	
29	NA0	

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

01 02 03	TPLM	<p>Определяет операцию с регистром инструкций <b>IR</b>, возможна произвольная комбинация операций:</p> <ul style="list-style-type: none"><li>• запуск чтения кода следующей инструкции на шине МПИ</li><li>• ожидание завершения чтения и записи прочитанного значения в регистр <b>IR</b></li><li>• запуск опроса блока прерываний и получения результата в <b>MCIR</b></li></ul> <table><tr><td></td><td>MCIR</td><td>Wait</td><td>Read</td><td>Примечание</td></tr><tr><td>000</td><td>нет</td><td>нет</td><td>нет</td><td>нет операции</td></tr><tr><td>001</td><td>нет</td><td>да</td><td>нет</td><td>ожидание завершения чтения кода следующей инструкции в <b>IR</b>, при этом текущая инструкция RTT</td></tr><tr><td>010</td><td>нет</td><td>нет</td><td>нет</td><td>нет операции</td></tr><tr><td>011</td><td>да</td><td>да</td><td>да</td><td>запуск чтения кода следующей инструкции по шине МПИ (если не было запущено ранее) с ожиданием завершения, одновременный запуск проверки состояния блока обработки прерываний</td></tr><tr><td>100</td><td>нет</td><td>нет</td><td>да</td><td>ранняя предвыборка кода следующей инструкции, чтение на шине МПИ запускается, но его завершение не ожидается в данном микроцикле</td></tr><tr><td>101</td><td>да</td><td>нет</td><td>нет</td><td>только проверка наличия прерываний, используется инструкцией WAIT</td></tr><tr><td>110</td><td>нет</td><td>нет</td><td>нет</td><td>нет операции</td></tr><tr><td>111</td><td>да</td><td>да</td><td>нет</td><td>ожидание завершения чтения кода следующей инструкции в <b>IR</b>, при этом текущая инструкция не RTT</td></tr></table>		MCIR	Wait	Read	Примечание	000	нет	нет	нет	нет операции	001	нет	да	нет	ожидание завершения чтения кода следующей инструкции в <b>IR</b> , при этом текущая инструкция RTT	010	нет	нет	нет	нет операции	011	да	да	да	запуск чтения кода следующей инструкции по шине МПИ (если не было запущено ранее) с ожиданием завершения, одновременный запуск проверки состояния блока обработки прерываний	100	нет	нет	да	ранняя предвыборка кода следующей инструкции, чтение на шине МПИ запускается, но его завершение не ожидается в данном микроцикле	101	да	нет	нет	только проверка наличия прерываний, используется инструкцией WAIT	110	нет	нет	нет	нет операции	111	да	да	нет	ожидание завершения чтения кода следующей инструкции в <b>IR</b> , при этом текущая инструкция не RTT
	MCIR	Wait	Read	Примечание																																											
000	нет	нет	нет	нет операции																																											
001	нет	да	нет	ожидание завершения чтения кода следующей инструкции в <b>IR</b> , при этом текущая инструкция RTT																																											
010	нет	нет	нет	нет операции																																											
011	да	да	да	запуск чтения кода следующей инструкции по шине МПИ (если не было запущено ранее) с ожиданием завершения, одновременный запуск проверки состояния блока обработки прерываний																																											
100	нет	нет	да	ранняя предвыборка кода следующей инструкции, чтение на шине МПИ запускается, но его завершение не ожидается в данном микроцикле																																											
101	да	нет	нет	только проверка наличия прерываний, используется инструкцией WAIT																																											
110	нет	нет	нет	нет операции																																											
111	да	да	нет	ожидание завершения чтения кода следующей инструкции в <b>IR</b> , при этом текущая инструкция не RTT																																											
06 07 08	QBUS	<p>Флаг байтовой операции на шине МПИ Флаг операции записи на шине МПИ Флаг операции чтения на шине МПИ</p> <table><tr><td>00x</td><td>нет операции</td></tr><tr><td>010</td><td>запись слова</td></tr><tr><td>011</td><td>запись байта</td></tr><tr><td>100</td><td>чтение слова</td></tr><tr><td>101</td><td>чтение байта (читается слово)</td></tr><tr><td>110</td><td>чтение-модификация-запись слова</td></tr><tr><td>111</td><td>чтение-модификация-запись байта</td></tr></table>	00x	нет операции	010	запись слова	011	запись байта	100	чтение слова	101	чтение байта (читается слово)	110	чтение-модификация-запись слова	111	чтение-модификация-запись байта																															
00x	нет операции																																														
010	запись слова																																														
011	запись байта																																														
100	чтение слова																																														
101	чтение байта (читается слово)																																														
110	чтение-модификация-запись слова																																														
111	чтение-модификация-запись байта																																														



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

4 21 22	PLOP	Управление записью арифметических флагов <b>PSW</b> из результирующих флагов АЛУ
10	Не ожидать данные чтения на шине МПИ	
11 13	PLX	Данные флаги являются многофункциональными, определяют тип операнда на шине <b>Y</b> , разрешают запись специального управляющего вектора, определяют тип операции АЛУ
12	UOP	<p>Неопределенная операция, при возникновении этого активного флага микропрограммный автомат переходит на адрес 01, все операции записи блокируются, устанавливается флаг запроса на исключение Undefined Opcode. Таким образом прекращается исполнение текущей последовательности микроинструкций и происходит переход на начальный анализ исключений. Также установка данного флага вызывает безусловный запрос на заполнение регистра <b>MCIR</b> блоком прерываний, в итоге процессор начнет обрабатывать исключение по вектору 000010<sub>8</sub> (если нет более приоритетных исключений). Флаг генерируется на всех недопустимых адресах, а также как результат декодирования кода инструкции в регистре <b>IR</b>.</p> <p>Флаг также блокирует на протяжении текущего микроцикла любые записи в регистры, слово состояния, и запуск транзакций на шине МПИ - это позволяет избежать разрушения состояния микропроцессора при неопределенной операции</p>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

13	ОП0 ОП1 ОП2 ОП3	Код операции АЛУ или выбор типа операнда на шине Y:			
14		000х, 0011	X + Y		
16		0010	X & ~Y		
17		010х, 0111	Y - X		
		0110	X ^ Y		
		1000	Y		
		10х1	X		
		1010	X   Y		
		110х, 1111	X - Y		
		1110	X & Y		
11, 13 и 14 разряды определяют тип операнда на шине Y - регистр, константа или вектор, а также разрешают запись специального управляющего вектора					
14	13	11	Трансляция на шину Y		
0	0	0	R0-R13, PSW, Q (индекс в разрядах 28-25)		
0	0	1	константа ([28:25] != 0010 <sub>2</sub> ) или вектор ([28:25] == 0010 <sub>2</sub> )		
0	1	0	константа ([28:25] != 0010 <sub>2</sub> ) или вектор ([28:25] == 0010 <sub>2</sub> )		
0	1	1	константа ([28:25] != 0010 <sub>2</sub> ) или вектор ([28:25] == 0010 <sub>2</sub> )		
1	0	0	R0-R13, PSW, Q (индекс в разрядах 28-25)		
1	0	1	константа ([28:25] != 0010 <sub>2</sub> ) или вектор ([28:25] == 0010 <sub>2</sub> )		
1	1	0	R12 (предположительно не используется)		
1	1	1	R12   константа/вектор (предположительно не используется)		
18	Признак словной операции : 0 - операция с байтом, 1 - операция со словом				
20	Фиксация - записать результат и флаги операции АЛУ в регистры POH/PSW				
23	Не ожидать освобождения регистра А или завершения записи на шине МПИ				

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

25

~Y3

26

~Y2

27

~Y1

28

~Y0

Данное поле имеет несколько значений:

Индекс регистра, определяет что будет выставлено на входную шину **Y** АЛУ блоком регистров. Если биты данного поля инвертировать и взять в обратном порядке, то получится индекс регистра, совпадающий (для регистров 0-7) с регистровым индексом в полях процессорных инструкций.

0000	<b>R0</b>	0100	<b>R4</b>	1000	<b>R8</b>	1100	<b>R12</b>
0001	<b>R1</b>	0101	<b>R5</b>	1001	<b>R9</b>	1101	<b>R13</b>
0010	<b>R2</b>	0110	<b>R6</b>	1010	<b>R10</b>	1110	<b>PSW</b>
0011	<b>R3</b>	0111	<b>R7</b>	1011	<b>R11</b>	1111	<b>Q</b>

Также поле отвечает за выбор константы, выдаваемой на шину **Y** как аргумент АЛУ (см. описание поля OP13-OP17):

0000	<b>IR</b> [3:0] - SE <sub>x</sub> /CL <sub>x</sub>	1000	<b>IR</b> [7:0]<<1 со знаком
0001	000340 <sub>8</sub>	1001	100000 <sub>8</sub>
0010	<b>VSEL</b> table	1010	177676 <sub>8</sub>
0011	000002 <sub>8</sub>	1011	000020 <sub>8</sub>
0100	<b>IR</b> [5:0]<<1 - SOB	1100	флаг переноса <b>C</b>
0101	177716 <sub>8</sub>	1101	177400 <sub>8</sub>
0110	177777 <sub>8</sub>	1110	000010 <sub>8</sub>
0111	000001 <sub>8</sub>	1111	000000 <sub>8</sub>

Также поле определяет режим сдвигателя на выходе АЛУ для одноадресных операций сдвига, если разряды 13 и 14 оба единичные, то разряды 27-25 определяют режим сдвига АЛУ и работу схемы обмена байт на входе **X** АЛУ:

000	влево, в младший бит вдвигается 0, ASL
001	влево, в младший бит вдвигается 0, ASL
010	влево, в младший бит вдвигается <b>C</b> , ROL
011	нет сдвига, перестановка байт на входе АЛУ с шины <b>X</b>
100	вправо, старший бит копируется, ASR
101	вправо, в старший бит вдвигается перенос из старшего разряда АЛУ в текущей операции
110	вправо, в старший бит вдвигается <b>C</b> , ROR
111	нет сдвига

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

30	~X3	Индекс аргумента <b>X</b> (биты инвертированы и порядок переставлен для совпадения с индексов регистра в поле процессорной инструкции), определяет что будет выставлено на общую шину <b>X</b> АЛУ блоком регистров:																																								
31	~X2																																									
32	~X1																																									
33	~X0																																									
			<table><tr><td>0000</td><td><b>R0</b></td><td>0100</td><td><b>R4</b></td><td>1000</td><td><b>R8</b></td><td>1100</td><td><b>R12</b></td></tr><tr><td>0001</td><td><b>R1</b></td><td>0101</td><td><b>R5</b></td><td>1001</td><td><b>R9</b></td><td>1101</td><td><b>R13</b></td></tr><tr><td>0010</td><td><b>R2</b></td><td>0110</td><td><b>R6</b></td><td>1010</td><td><b>R10</b></td><td>1110</td><td><b>PSW</b></td></tr><tr><td>0011</td><td><b>R3</b></td><td>0111</td><td>и</td><td>1011</td><td><b>R11</b></td><td>1111</td><td><b>Q</b></td></tr></table>								0000	<b>R0</b>	0100	<b>R4</b>	1000	<b>R8</b>	1100	<b>R12</b>	0001	<b>R1</b>	0101	<b>R5</b>	1001	<b>R9</b>	1101	<b>R13</b>	0010	<b>R2</b>	0110	<b>R6</b>	1010	<b>R10</b>	1110	<b>PSW</b>	0011	<b>R3</b>	0111	и	1011	<b>R11</b>	1111	<b>Q</b>
0000	<b>R0</b>	0100	<b>R4</b>	1000	<b>R8</b>	1100	<b>R12</b>																																			
0001	<b>R1</b>	0101	<b>R5</b>	1001	<b>R9</b>	1101	<b>R13</b>																																			
0010	<b>R2</b>	0110	<b>R6</b>	1010	<b>R10</b>	1110	<b>PSW</b>																																			
0011	<b>R3</b>	0111	и	1011	<b>R11</b>	1111	<b>Q</b>																																			

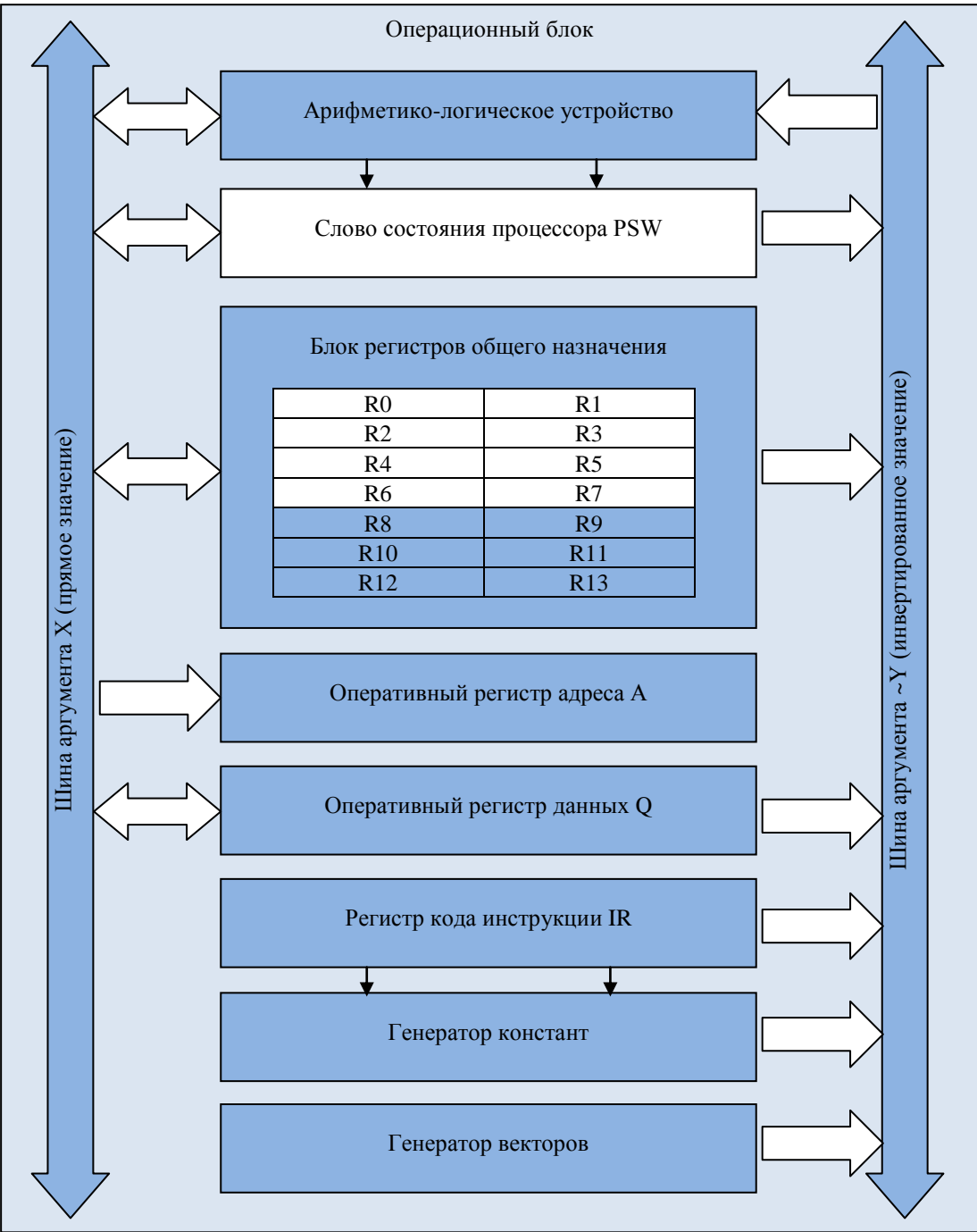
Если разряды 11 и 13 оба единичные, то происходит запись выходов логической матрицы в специальный управляющий вектор. Специальный управляющий вектор - это динамический набор стробов, управляющий работой различных блоков процессора. Назначение разрядов логической матрицы, соответствующих стробам:

7	0 - установить флаг <b>WCPU</b> , при опросе блока прерываний микропрограммный автомат будет остановлен и процессор перейдет в режим ожидания внешних прерываний, используется при выполнении инструкции WAIT	
27	0 - сбросить флаг <b>WCPU</b> и запустить на шине МПИ транзакцию подтверждения прерывания, процессор выходит из режима ожидания внешнего прерывания	
10	0 - активировать внешний сигнал <b>nINIT</b> . Значение внешнего выходного сигнала <b>nINIT</b> хранится в однобитовом регистре, который устанавливается в активное значение при записи управляющего вектора с нулевым 10-ым разрядом и устанавливается в неактивное при записи вектора с нулевым 23-им разрядом, используется при исполнении инструкции RESET	
23	0 - деактивировать внешний сигнал <b>nINIT</b> . Значение внешнего выходного сигнала <b>nINIT</b> хранится в однобитовом регистре, который устанавливается в активное значение при записи управляющего вектора с нулевым 10-ым разрядом и устанавливается в неактивное при записи вектора с нулевым 23-им разрядом, используется при исполнении инструкции RESET	
25	0 - установить флаг <b>HLT</b> в регистре управления режимом ( $177700_8 + \langle \text{номер процессора} \rangle * 20_8$ ), практически эта возможность рабочей версией микропрограммы не используется, процессор остановится	
26	0 – установить флаг <b>ER2</b> в регистре ошибок ( $177704_8 + \langle \text{номер процессора} \rangle * 20_8$ ), практически эта возможность рабочей версией микропрограммы не используется	
28	0 – установить флаг <b>ER3</b> в регистре ошибок ( $177704_8 + \langle \text{номер процессора} \rangle * 20_8$ ), практически эта возможность рабочей версией микропрограммы не используется	
30	0 – установить флаг <b>ER7</b> в регистре ошибок ( $177704_8 + \langle \text{номер процессора} \rangle * 20_8$ ), практически эта возможность рабочей версией микропрограммы не используется	
14 16 17	~MCIR2 MCIR1 ~MCIR0	Значение для непосредственной записи регистра статуса прерываний <b>MCIR</b> (см. описание блока прерываний)
18 20 21 22	VSEL3 ~VSEL2 ~VSEL1 ~VSEL0	Значение для непосредственной записи регистра выбора константы вектора прерываний (см. описание блока прерываний)

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

### 7.2 Операционный блок

Структурная схема операционного блока приведена на диаграмме:



Компоненты, выделенные светлым фоном, являются программно доступными - слово состояния процессора **PSW** и регистры общего назначения **R0-R7**.

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Арифметико-логическое устройство принимает на входах с внутренних шин операционного блока **X** и **~Y** два аргумента, выполняет над ними логическую или арифметическую операцию и выдает результат обратно на шину **X** (получение аргумента с шины **X** и выдача на нее результата разнесены по тактам, внутри АЛУ имеется регистр для фиксации аргумента **X**). Также по результатам операции вырабатываются флаги, которые могут быть опционально переписаны в слово состояния процессора **PSW**. Флаг переноса **C** может быть записан отдельно от флагов **N**, **Z**, **V**, которые всегда записываются единой группой. В процессоре 1801BM1 все арифметические операции, включая авто-инкремент счетчика команд **PC**, исполняются исключительно единственным основным АЛУ процессорного блока. На входе **X** АЛУ имеется схема обмена байт, которая опционально может переставить два байта входного аргумента местами.

Слово состояния процессора **PSW** содержит арифметические и управляющие флаги. Арифметические флаги устанавливаются согласно результатам выполнения операции АЛУ, управляющие флаги могут быть только непосредственно записаны как результат прямой записи слова в **PSW** с шины **X**. При записи слова состояния с шины **X** запись в старший байт может быть замаскирована, предохраняя старший байт от изменения. Например, инструкция **MTPS** не изменяет старший байт **PSW**. При аппаратном сбросе процессора слово состояния обнуляется, но процедура начального старта, реализованная в микрокоде, записывает в **PSW** значение 000340<sub>8</sub>.

Описание флагов слова состояния процессора **PSW** приведено в таблице:

Бит	Имя	Описание
0	<b>C</b>	Флаг переноса, устанавливается, если был перенос из старшего разряда (15-го при словных операциях и 7-го при байтовых). Операция вычитания реализуется как сложение с инвертированным вторым операндом плюс один, поэтому при вычитании флаг переноса устанавливается если не было займа в старший разряд (не аппаратной инверсии флага переноса при вычитании). При выполнении сдвигов действуют правила описанные в соответствующих операциях
1	<b>V</b>	Флаг арифметического переполнения, при сложении вычисляется как исключающее ИЛИ переносов из старшего и предстаршего разрядов. Обычно сигнализирует о потере знака результата, используется в знаковой арифметике
2	<b>Z</b>	Флаг равенства результата нулю, устанавливается, если все биты результата ([15:0] при словных операциях и [7:0] при байтовых)
3	<b>N</b>	Флаг знака результата, равен старшему разряду (15-му для словных операций и 7-му для байтовых)
4	<b>T</b>	Флаг ловушки пошаговой отладки, исключение зависит от режима работы процессора - например, сам по себе установленный бит <b>T</b> в режиме ожидания прерывания по команде <b>WAIT</b> не вызывает исключения. При выполнении инструкции <b>WAIT</b> незамаскированные прерывания <b>IRQ1</b> , <b>TVE</b> , <b>IRQ2</b> , <b>IRQ3</b> , <b>VIRQ</b> имеют приоритет над исключением отладки. В остальных случаях установленный бит <b>T</b> вызывает программное исключение по вектору 000014 <sub>8</sub> . Инструкция <b>MTPS</b> не изменяет данный бит.
5,6	-	Резервные биты, могут быть записаны и прочитаны, но не оказывают никакого влияния
7	<b>I</b>	Единичное значение бита <b>I</b> маскирует следующие прерывания: - прерывание от BE-таймера (только K1801BM1Г) - <b>nIRQ2</b> - <b>nIRQ3</b> - векторное прерывание <b>nVIRQ</b>
8	<b>PA0</b>	Младший бит номера процессора в системе, инверсное значение входа <b>nPA0</b> . Данный

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

		бит может быть только прочитан, не изменяется при записи
9	<b>PA1</b>	Старший бит номера процессора в системе, инверсное значение входа <b>nPA1</b> . Данный бит может быть только прочитан, не изменяется при записи
10	<b>PSW10</b>	<p>Модифицирует обработку исключения зависания шины, если данный бит ненулевой, то исключение зависания будет обрабатываться в пультовом режиме по вектору 160002<sub>8</sub></p> <p>Единичное значение бита <b>PSW10</b> также маскирует следующие прерывания:</p> <ul style="list-style-type: none"> <li>- по пропаданию питания (срез на входе <b>nACLO</b>)</li> <li>- пультовое прерывание <b>nIRQ1</b></li> <li>- прерывание от BE-таймера (только K1801BM1Г)</li> <li>- <b>nIRQ2</b></li> <li>- <b>nIRQ3</b></li> <li>- векторное прерывание <b>nVIRQ</b></li> </ul> <p>Бит всегда сбрасывается при входе в процедуру обработки прерывания или исключения пользовательского режима, значение бита, читаемое из вектора игнорируется. Поскольку данный бит маскирует асинхронные прерывания пользовательского режима, то его сброс может произойти только по синхронным пользовательским исключениям (ошибки или инструкции вызова исключений).</p> <p>Бит также всегда сбрасывается при выполнении инструкций RTT и RTI, значение загружаемое из стека игнорируется. Фактически данный бит может быть установлен только при входе в обработчик прерывания или исключения пультового режима (хорошо подходит инструкция HALT), а также при возврате из него инструкциями STEP и START. Инструкция MTPS не изменяет данный бит.</p>
11	<b>PSW11</b>	<p>Модифицирует обработку исключения зависания шины, если данный бит ненулевой, то исключение зависания будет обрабатываться в пультовом режиме по вектору 160002<sub>8</sub>, иначе обработка по вектору 000004<sub>8</sub> без перехода в пультовой режим</p> <p>Единичное значение бита <b>PSW11</b> также запрещает пультовое прерывание <b>nIRQ1</b>.</p> <p>Бит всегда сбрасывается при входе в процедуру обработки прерывания или исключения пользовательского режима, значение бита, читаемое из вектора игнорируется.</p> <p>Бит также всегда сбрасывается при выполнении инструкций RTT и RTI, значение загружаемое из стека игнорируется. Фактически данный бит может быть установлен только при входе в обработчик прерывания или исключения пультового режима (хорошо подходит инструкция HALT), а также при возврате из него инструкциями STEP и START. Инструкция MTPS не изменяет данный бит</p>
12-15	-	Резервные биты, могут быть записаны и прочитаны, но не оказывают никакого влияния. Обнуляются при входе в обработчик прерывания или исключения пользовательского режима – значения читаемые из вектора прерывания или исключения игнорируются. Инструкция MTPS не изменяет данные биты

Блок регистров общего назначения содержит 14 шестнадцатиразрядных регистров. При выполнении операций возможна отдельная запись только младшего байта (8 младших бит), так и запись всего шестнадцатиразрядного слова целиком, включая старший байт. Запись только младшего байта в регистры используется при байтовых операциях (за исключением MOVB, когда в старший байт регистра записывается знаковое расширение младшего байта). Блок регистров общего назначения является двухпортовым ОЗУ и может обеспечить одновременную выдачу двух разных регистров на шины **X** и **Y**.

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Регистры **R8-R13** программно недоступны и используются микропрограммным автоматом для собственных нужд (хранение адресов операндов, временных переменных и так далее). Регистр **R7** выполняет функцию программного счетчика **PC** и используется для адресации потока выполняемых процессорных инструкций. Регистр **R6** выполняет функцию указателя стека **SP**, и используется в процедурах исключений и прерываний.

Оперативный регистр адреса **A** предназначен исключительно для записи по шине **X** адреса запрашиваемой транзакции МПИ. Регистр **A** может быть записан в различных фазах цикла микропрограммного автомата – как до выполнения операции АЛУ, так и после. Например, при запуске извлечения кода очередной исполняемой процессорной инструкции по адресу в счетчике команд **PC**, в регистр адреса **A** записывается значение **PC** в начале цикла, до выполнения операции автоинкремента **PC+2**. После записи значения адреса регистр переходит в занятое состояние и освобождается после выполнения фазы трансляции адреса в транзакции на шине МПИ. Далее автомат транзакций МПИ продолжает обработку транзакции, а регистр адреса готов к записи нового значения для следующей транзакции. Таким образом, процессор 1801BM1 способен конвейеризировать до двух транзакций МПИ. При записи в регистр адреса также фиксируется тип транзакции, для сохранения типа также предусмотрен буфер хранения до двух транзакций МПИ. При записи регистра адреса процессор выполняет следующие действия:

- Ожидает завершения предыдущих транзакций МПИ, если таковые есть
- Выставляет запрос на захват шины и ожидает ее предоставления
- Начинает фазу трансляции адреса (активный выход **nSYNC**)
- После завершения фазы трансляции адреса переходит к фазе чтения данных для транзакций чтения и чтения-модификации-записи (активный выход **nDIN**)
- Если транзакция подразумевает запись, то ожидается запись регистра **Q**, таковая может быть выполнена еще до записи регистра **A** или после, в транзакция модификации ожидается запись в регистр **Q** результатов модификации данных со стороны АЛУ
- Ожидание завершения транзакций не производится, ожидание данных происходит на уровне микропрограммного автомата

Оперативный регистр данных **Q** служит для сохранения данных прочитанных в результате транзакций чтения по шине МПИ, а также для сохранения данных, записываемых по шине **X** со стороны АЛУ, предназначенных для последующей записи по шине МПИ во внешние устройства или память.

Регистр кода инструкции **IR** предназначен для хранения кода текущей процессорной инструкции, обрабатываемой в данный момент микропрограммным автоматом. Также имеется теневой регистр для хранения кода следующей инструкции, полученной в результате предварительной выборки (чтение кода инструкции запущенное заранее, еще в ходе обработки текущей инструкции).

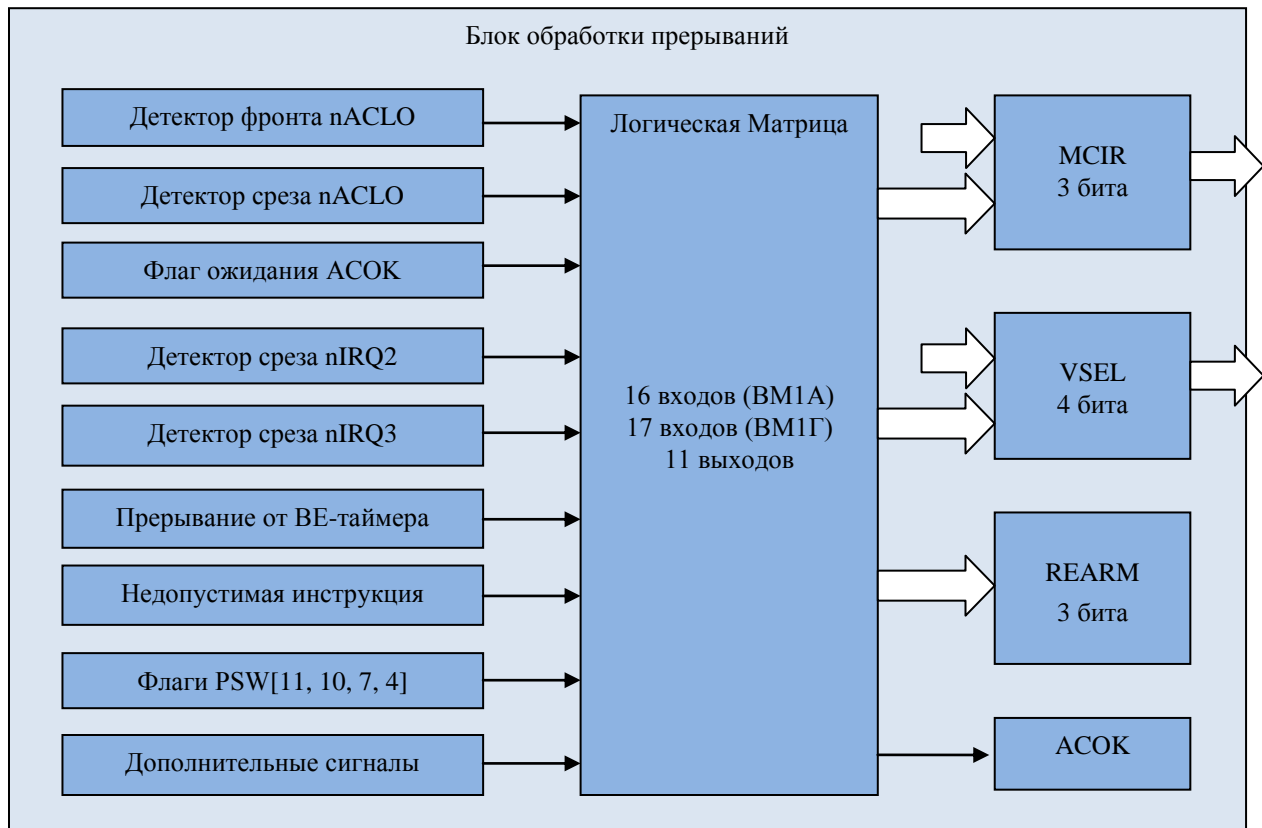
Генераторы констант и векторов выдают на шину **Y** одно из постоянных значений из фиксированного набора. В набор входят значения векторов исключений, некоторые константы для вычислений - -1, 0, 1, 2, а также фиксированные адреса регистров и указателя стека пультового режима.

Во многих документах и обсуждениях, касающихся 1801BM1, встречается упоминание так называемого “пультового режима”. На самом деле внутри процессора нет никаких специальных схем и состояний для специального пультового режима. Вместо этого предусмотрен условный бит управления режимом, вынесенный во внешние схемы - бит 3 регистра 177716<sub>8</sub> (этот адрес НЕ зависит от физического номера процессора, одинаков для всех процессоров в многопроцессорной системе). При входе в некоторые исключения микропрограмма вызывает обращение к регистру 177716<sub>8</sub> и установку бита, такие исключения называются исключениями пультового режима. Специальные инструкции **START** и **STEP** производят возврат из таких исключений (являются аналогами инструкций **RTI** и **RTT**), при этом также производится обращение к регистру 177716<sub>8</sub> и сброс бита 3.



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

### 7.3 Блок обработки прерываний



Блок обработки прерываний принимает и детектирует запросы на прерывания и исключения от различных источников, упорядочивает их по приоритету, выполняет маскирование и передает информацию о имеющемся наиболее приоритетном незамаскированном активном запросе для блока микропрограммного управления.

Запросы на прерывание или исключение поступают на детектор. Детектор производит непрерывную проверку линии запроса и при обнаружении события фиксирует соответствующий флаг. Например, детектор среза **nACLO** будет выставлять активный флаг при обнаружении перехода входа **nACLO** в низкий уровень. Флаг останется выставленным до события аппаратного сброса процессора или до тех пор пока информация об этом запросе не будет передана и обработана блоком микропрограммного управления.

Обнаруженные и зафиксированные запросы поступают на логическую матрицу, которая также обрабатывает функции маскирования и упорядочивания запросов по приоритетам. Матрица имеет 19 входов в K1801BM1A и 20 входов в K1801BM1Г (в последнем дополнительный вход является запросом прерывания от ВЕ-таймера, в остальном матрицы одинаковы) и 11 выходов. 3 входа являются резервными и не используются.

Выходные сигналы матрицы фиксируются в трех отдельных регистрах и одном флаге:

- **MCIR** - 3-битовый регистр флагов наличия прерываний, его выход напрямую подается на матрицу блока микропрограммного управления. Этот регистр может быть записан как с выходов матрицы блока прерываний, так и данными со стороны микропрограммного блока. Запись этого регистра со стороны блока микропрограммного управления используется при отработке программных исключений, вызываемых инструкциями типа HALT, EMT и т.д.

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

- **VSEL** - 4-битовый регистр индекса вектора прерывания. Подается на вход таблицы генератора констант, может быть записан как с выходов матрицы блока прерываний, так и данными со стороны микропрограммного блока.
- **REARM** - 3-битовый регистр, определяет индекс детектора события, флаг которого должен быть сброшен, используется исключительно внутри блока прерываний для сигнализации детектору события что флаг обработан и надо реактивироваться для обнаружения следующего события.
- **АСОК** - флаг нормального старта. Этот флаг обнуляется при аппаратном сбросе микропроцессора, и блок прерываний переходит в ожидание события фронта на **nACLO**. При этом блоку микропрограммного управления сообщается специальный код, и тот находится в пустом начальном цикле, ожидая начального старта.

Блок микропрограммного управления в первом такте декодирования инструкции выполняет транзакцию опроса блока прерываний. При этой транзакции опроса блок прерываний записывает информацию о имеющихся запросах в регистр **MCIR**, также в регистр **VSEL** записывается индекс требуемого вектора прерывания. Если на выход для обработки передан запрос с детектора события, то соответствующий детектор сбрасывается. Например, детектор среза **nIRQ3** обнаружил событие среза, выставил соответствующий флаг запроса, когда этот флаг будет обработан (что необязательно происходит в первом же цикле опроса, так как запрос может быть замаскирован флагами в слове состояния **PSW** или могут иметься более приоритетные запросы), то в регистр **REARM** будет записан код для сброса именно детектора **nIRQ3** и детектор инициализируется и перейдет в ожидание нового события среза по этой линии.

Значения регистра **MCIR** и реакция блока микропрограммного управления на них

000 <sub>2</sub>	Ожидание прерывания. Блок микропрограммного управления постоянно выполняет опрос блока прерывания в цикле и никаких других операций не производится. Это состояние используется для ожидания начального пуска - флаг <b>АСОК</b> сброшен, блок прерывания ожидает фронта <b>nACLO</b> . И также при выполнении инструкции <b>WAIT</b> - постоянно ожидается поступление прерывания.
001 <sub>2</sub>	Нет активных незамаскированных запросов на прерывание или исключение
01x <sub>2</sub>	Переход в "пультовый" режим (по инструкции <b>HALT</b> , низкому уровню на входе <b>nIRQ1</b> , двойной ошибке). Внутри процессора нет специального пультового режима, бит управления режимом вынесен во внешние схемы - бит 3 регистра 177716 <sub>8</sub> (адрес НЕ зависит от физического номера процессора). При входе в режим осуществляется чтение-модификация-запись внешнего регистра 177716 <sub>8</sub> с установкой маски 10 <sub>8</sub> . Установленный младший бит указывает, что не требуется выполнять коррекцию <b>PC</b> (уменьшение на 2) перед сохранением в стеке
10x <sub>2</sub>	Требуется обработка прерывания или исключения пользовательского режима, установленный младший бит указывает, что не требуется выполнять коррекцию <b>PC</b> (уменьшение на 2) перед сохранением в стеке
11x <sub>2</sub>	Выполнить последовательность начального старта (прочитать регистр начального старта, инициализировать <b>PC</b> и <b>PSW</b> )

Значения регистра **VSEL** и соответствующие индексам значения векторов

00 <sub>8</sub>	160006 <sub>8</sub> - исключение по двойной ошибке шины
-----------------	---

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

01 <sub>8</sub>	000020 <sub>8</sub> - инструкция <b>IOT</b> (это значение индекса вектора никогда не генерируется блоком прерываний, а записывается со стороны микропрограммного блока при декодировании инструкции)
02 <sub>8</sub>	000010 <sub>8</sub> - зарезервированный код инструкции
03 <sub>8</sub>	000014 <sub>8</sub> - установленный бит <b>T</b> в слове состояния процессора <b>PSW</b>
04 <sub>8</sub>	000004 <sub>8</sub> - тайм-аут (зависания) внешней шины
05 <sub>8</sub>	Адрес начального пуска, значение вектора зависит от физического номера процессора 177716 <sub>8</sub> - процессор номер 0 177736 <sub>8</sub> - процессор номер 1 177756 <sub>8</sub> - процессор номер 2 177776 <sub>8</sub> - процессор номер 3
06 <sub>8</sub>	000030 <sub>8</sub> - инструкция <b>EMT</b> (это значение индекса вектора никогда не генерируется блоком прерываний, а записывается в регистр <b>MCIR</b> со стороны микропрограммного блока при декодировании инструкции)
07 <sub>8</sub>	160012 <sub>8</sub> - тайм-аут шины при приеме вектора прерывания
10 <sub>8</sub>	000270 <sub>8</sub> - обнаружение спада на входе <b>nIRQ3</b> , или BE-таймер (только K1801BM1Г)
11 <sub>8</sub>	000024 <sub>8</sub> - обнаружение спада на входе <b>nACLO</b>
12 <sub>8</sub>	000100 <sub>8</sub> - обнаружение спада на входе <b>nIRQ2</b>
13 <sub>8</sub>	160002 <sub>8</sub> - низкий уровень на входе <b>nIRQ1</b> или инструкция <b>HALT</b>
14 <sub>8</sub>	000034 <sub>8</sub> - инструкция <b>TRAP</b> (это значение индекса вектора никогда не генерируется блоком прерываний, а записывается со стороны микропрограммного блока при декодировании инструкции)
15 <sub>8</sub>	Получить вектор прерывания от внешнего устройства (это значение индекса вектора никогда не генерируется блоком прерываний)
16 <sub>8</sub>	Получить вектор прерывания из регистра вектора прерывания рестарта 1777x2 <sub>8</sub> (это значение индекса вектора никогда не генерируется блоком прерываний, так как эта возможность не реализована в микропроцессоре, вероятно рудиментарный остаток от K1801BE1)
17 <sub>8</sub>	000000 <sub>8</sub> - неиспользуемый вектор

#### Значения регистра **REARM** и соответствие детекторам

000 <sub>2</sub>	Флаг неопределенной инструкции <b>UERR</b> . этот флаг устанавливается блоком микропрограммного управления при обнаружении ошибки декодирования кодов операций, после установки флага ошибки блок микропрограммного управления сбрасывается и переходит к моменту опроса состояния блока прерывания. Флаг неопределенной инструкции является приоритетным и немаскируемым и блок прерывания выставляет соответствующий код, вызывающий переход блока микропрограмм к процедуре обработки исключения. После передачи кода блок прерываний очищает этот флаг. Также этот индекс сбрасывает флаги таймаутов по внешней шине для данных и вектора прерываний - <b>QBTO</b> и <b>IATO</b> .
001 <sub>2</sub>	Флаг обнаружения спада на входе <b>nIRQ3</b>
011 <sub>2</sub>	Флаг обнаружения спада на входе <b>nIRQ2</b>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

100 <sub>2</sub>	Флаги обнаружения фронта и спада на входе <b>nACLO</b>
101 <sub>2</sub>	Флаг прерывания от ВЕ-таймера, устанавливается таймером, сбрасывается при обработке прерывания, есть только в K1801BM1Г

Таблица входов матрицы и формируемые выходные значения

Вход	Описание	MCIR	VSEL	REA
<b>АСОК</b>	Флаг ожидания начального пуска, при аппаратном сбросе микропроцессора этот флаг сбрасывается и будет установлен только после возникновения события фронта на входе <b>nACLO</b> и обработки этого события матрицей (флаг устанавливается значением с выхода матрицы, после первичной установки будет оставаться установленным постоянно). После обработки начального пуска флаг устанавливается и в дальнейшей работе влияния на матрицу не оказывает			
	<b>АСОК</b> =0, <b>ACIN</b> =0 (нет фронта <b>nACLO</b> )	000 <sub>2</sub>	11 <sub>8</sub>	111 <sub>2</sub>
	<b>АСОК</b> =0, <b>ACIN</b> =1 (есть фронт <b>nACLO</b> )	111 <sub>2</sub>	5 <sub>8</sub>	100 <sub>2</sub>
	<b>АСОК</b> =1 - нормальная работа матрицы приоритетов прерываний			
<b>ACIN</b>	Детектор фронта на входе <b>nACLO</b> , этот вход работает только при ожидании начального пуска, в дальнейшем не оказывает влияния на функционирование блока обработки прерываний	-	-	-
<b>IATO</b>	Тайм-аут получения вектора прерывания по внешней шине, всегда устанавливается вместе с флагом <b>QBTO</b> , сбрасывается после передачи кода блоку микропрограмм	011 <sub>2</sub>	7 <sub>8</sub>	000 <sub>2</sub>
<b>DBLE</b>	Флаг двойной ошибки. Если произошел тайм-аут шины, или исключение неопределенной инструкции, и при обработке исключения снова возник тайм-аут или неопределенная инструкция, то этот флаг устанавливается. Сброс флага происходит при загрузке кода новой инструкции в регистр инструкции. На практике может возникать только двойной тайм-аут (повторная неопределенная инструкция может возникнуть только при нарушении работы микрокода), поэтому этот флаг всегда сопровождается флагом <b>QBTO</b>	011 <sub>2</sub>	0 <sub>8</sub>	000 <sub>2</sub>
<b>QBTO</b>	тайм-аут внешней шины МПИ. При активации стробов <b>nDIN</b> или <b>nDOUT</b> запускается специальный таймер, который тактируется от делителя <b>CLC/8</b> , если в ответ не поступит <b>nRPLY</b> , то после отсчета таймером 8 тиков устанавливается флаг <b>QBTO</b> . Делитель частоты /8 не сбрасываемый, поэтому фактический тайм-аут может составлять от 56 до 64 тактов процессорной частоты			
	<b>PSW10</b> =0, <b>PSW11</b> =0, исключение по вектору 000004 <sub>8</sub>	101 <sub>2</sub>	4 <sub>8</sub>	000 <sub>2</sub>
	<b>PSW10</b> =1, <b>PSW11</b> =0, исключение по вектору 160002 <sub>8</sub>	011 <sub>2</sub>	13 <sub>8</sub>	000 <sub>2</sub>
	<b>PSW10</b> =x, <b>PSW11</b> =1, исключение по вектору 160002 <sub>8</sub> Примечание: K1801BM1Г формирует значение <b>MCIR</b> =011 <sub>2</sub>	010 <sub>2</sub>	13 <sub>8</sub>	000 <sub>2</sub>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Вход	Описание	MCIR	VSEL	REA
<b>UERR</b>	Недопустимая инструкция, флаг устанавливается блоком микропрограммного управления при возникновении ошибки декодирования кода инструкции, загруженной в регистр инструкций. Флаг сбрасывается блоком прерываний при передаче кода старта обработки исключения блоку микропрограмм	101 <sub>2</sub>	2 <sub>8</sub>	000 <sub>2</sub>
<b>PSW4</b>	<b>T</b> -бит ловушки пошаговой отладки в слове состояния процессора <b>PSW</b> , прерывание зависит от режима работы процессора - например, сам по себе установленный бит <b>T</b> в режиме ожидания прерывания по команде <b>WAIT</b> не вызывает исключения. Ниже рассмотрены условия при <b>PSW11=0</b> , <b>PSW10=0</b> , <b>PSW7=0</b> , <b>PSW4=1</b> . То есть прерывания <b>IRQ1</b> , <b>TVE</b> , <b>IRQ2</b> , <b>IRQ3</b> , <b>VIRQ</b> дополнительно не замаскированы. Если установлены биты маски, то соответствующий бит запроса следует просто считать нулевым (замаскированным) в приведенных ниже выражениях. Запрос <b>TVE</b> присутствует только в K1801BM1Г, для K1801BM1А его следует считать неактивным (нулевым).			
	<b>WCPU=0</b> , активный режим, прерывание по вектору 000014 <sub>8</sub>	100 <sub>2</sub>	11 <sub>8</sub>	010 <sub>2</sub>
	<b>WCPU=1</b> , режим ожидания прерывания			
	<b>IRQ1=1</b> , <b>TVE=x</b> , <b>IRQ2=x</b> , <b>IRQ3=x</b> , <b>VIRQ=x</b> , переход в пульт	010 <sub>2</sub>	13 <sub>8</sub>	111 <sub>2</sub>
	<b>IRQ1=0</b> , <b>TVE=1</b> , <b>IRQ2=x</b> , <b>IRQ3=x</b> , <b>VIRQ=x</b> , прерывание таймера	100 <sub>2</sub>	10 <sub>8</sub>	101 <sub>2</sub>
	<b>IRQ1=0</b> , <b>TVE=0</b> , <b>IRQ2=1</b> , <b>IRQ3=x</b> , <b>VIRQ=x</b> , прерывание <b>nIRQ2</b>	100 <sub>2</sub>	12 <sub>8</sub>	011 <sub>2</sub>
	<b>IRQ1=0</b> , <b>TVE=0</b> , <b>IRQ2=0</b> , <b>IRQ3=1</b> , <b>VIRQ=x</b> , прерывание <b>nIRQ3</b>	100 <sub>2</sub>	10 <sub>8</sub>	001 <sub>2</sub>
	<b>IRQ1=0</b> , <b>TVE=0</b> , <b>IRQ2=0</b> , <b>IRQ3=0</b> , <b>VIRQ=1</b> , прерывание <b>nVIRQ</b>	100 <sub>2</sub>	14 <sub>8</sub>	010 <sub>2</sub>
	<b>IRQ1=0</b> , <b>TVE=0</b> , <b>IRQ2=0</b> , <b>IRQ3=0</b> , <b>VIRQ=0</b> , ожидание	000 <sub>2</sub>	13 <sub>8</sub>	111 <sub>2</sub>
<b>ACLO</b>	Обнаружен ниспадающий фронт на входе <b>nACLO</b> , соответствует событию раннего оповещения о пропадании питания.			
	<b>PSW10=1</b> - <b>ACLO</b> замаскирован	001 <sub>2</sub>	13 <sub>8</sub>	111 <sub>2</sub>
	<b>PSW10=0</b> , <b>PSW4=1</b> , <b>WCPU=0</b> - прерывание по биту <b>T</b> из активного режима, вектор 000014 <sub>8</sub>	100 <sub>2</sub>	3 <sub>8</sub>	010 <sub>2</sub>
	<b>PSW10=0</b> , <b>PSW4=0</b> или <b>WCPU=1</b> - прерывание по вектору 000024 <sub>8</sub> если нет <b>T</b> бита или спящий режим (режим ожидания прерывания по команде <b>WAIT</b> )	100 <sub>2</sub>	11 <sub>8</sub>	100 <sub>2</sub>
<b>IRQ1</b>	Прерывание "аппаратного" останова и перехода в пультовый режим, может быть замаскировано битами <b>PSW10</b> или <b>PSW11</b>	010 <sub>2</sub>	13 <sub>8</sub>	111 <sub>2</sub>
<b>TVE</b>	Прерывание от BE-таймера, детектор события достижения нуля. Имеет общий разделяемый с <b>IRQ3</b> вектор 000270 <sub>8</sub> , маскируется единичными битами <b>PSW10</b> , <b>PSW7</b> и нулевым битом <b>MON</b> в регистре управления таймером	100 <sub>2</sub>	10 <sub>8</sub>	101 <sub>2</sub>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Вход	Описание	MCIR	VSEL	REA
<b>IRQ2</b>	Прерывание по срезу на входе <b>nIRQ2</b> , маскируется битами <b>PSW10, PSW7</b>	100 <sub>2</sub>	12 <sub>8</sub>	011 <sub>2</sub>
<b>IRQ3</b>	Прерывание по срезу на входе <b>nIRQ3</b> , маскируется битами <b>PSW10, PSW7</b>	100 <sub>2</sub>	10 <sub>8</sub>	001 <sub>2</sub>
<b>VIRQ</b>	Векторное прерывание, запрос возникает по низкому уровню на входе nVIRQ, маскируется битами <b>PSW10, PSW7</b>	100 <sub>2</sub>	14 <sub>8</sub>	010 <sub>2</sub>
<b>PSW7</b>	Единичное значение бита <b>PSW7</b> маскирует следующие прерывания: - прерывание от BE-таймера (только K1801BM1Г) - <b>nIRQ2</b> - <b>nIRQ3</b> - векторное прерывание <b>nVIRQ</b>			
<b>PSW10</b>	Модифицирует обработку исключения зависания шины, если данный бит ненулевой, то исключение зависания будет обрабатываться в пультовом режиме по вектору 160002 <sub>8</sub> , иначе обработка по вектору 000004 <sub>8</sub> без перехода в пульт  Единичное значение бита <b>PSW10</b> также маскирует следующие прерывания: - по пропаданию питания (срез на входе <b>nACLO</b> ) - пультовое прерывание <b>nIRQ1</b> - прерывание от BE-таймера (только K1801BM1Г) - <b>nIRQ2</b> - <b>nIRQ3</b> - векторное прерывание <b>nVIRQ</b>  Также замаскированные прерывания не осуществляют выход из режима ожидания по инструкции WAIT  При выполнении инструкции WAIT блок микропрограммного управления продолжает опрос блока прерывания до появления незамаскированного события			
<b>PSW11</b>	Модифицирует обработку исключения зависания шины, если данный бит ненулевой, то исключение зависания будет обрабатываться в пультовом режиме по вектору 160002 <sub>8</sub> , иначе обработка по вектору 000004 <sub>8</sub> без перехода в пульт  Единичное значение бита <b>PSW11</b> также запрещает пультовое прерывание <b>nIRQ1</b> .			
<b>WCPU</b>	Признак режима ожидания внешнего прерывания, вход в этот режим осуществляется по инструкции WAIT, выход по незамаскированному внешнему прерыванию. Установленный бит Т не приводит к немедленному выходу из режима ожидания, если нет активных незамаскированных аппаратных прерываний.			

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Таблица прерываний и исключений, значения векторов, стека сохранения **PSW** и **PC** и зависимость от маски запрета, расположены в порядке приоритета

	Описание	Вектор	PSW/PC	Маска
1	Начальный пуск после аппаратного сброса (читается только стартовый адрес в старшем байте регистра, PSW всегда устанавливается в 340 <sub>8</sub> )	177716 <sub>8</sub> + N <sub>cpu</sub> *20 <sub>8</sub>	-	-
2	Зависание шины при получении вектора прерывания (в цикле <b>nIAKO</b> )	160012 <sub>8</sub>	177674 <sub>8</sub>	-
3	Двойное зависание шины (повторное при попытке обработать первое исключение по зависанию)	160006 <sub>8</sub>	177674 <sub>8</sub>	-
4	Зависание шины, обработка зависит от битов 10 и 11 в <b>PSW</b> : если <b>PSW11=0</b> , <b>PSW10=0</b> если хотя бы один бит <b>PSW10</b> и <b>PSW11</b> ненулевой	000004 <sub>8</sub> 160002 <sub>8</sub>	SP-4 177674 <sub>8</sub>	<b>PSW11</b> <b>PSW10</b>
5	Недопустимый код инструкции	000010 <sub>8</sub>	SP-4	-
6	<b>T</b> -бит, бит 4 в слове состояния процессора <b>PSW</b> . Если данный бит имеет единичное значение то в обычном режиме вызывает исключение по вектору 000014 <sub>8</sub> . В режиме ожидания по команде <b>WAIT</b> <b>T</b> -бит не вызывает немедленного исключения, процессор ожидает прерывание, любое незамаскированное прерывание <b>ACLO</b> , <b>IRQ1</b> , <b>TVE</b> , <b>IRQ2</b> , <b>IRQ3</b> , <b>VIRQ</b> вызывает выход из этого режима и переход по вектору, соответствующему активному прерыванию	000014 <sub>8</sub>	SP-4	<b>WCPU</b>
7	Пропадание питания - ранее оповещение, обнаружен ниспадающий фронт на входе <b>nACLO</b> .	000024 <sub>8</sub>	SP-4	<b>PSW10</b>
8	Прерывание "аппаратного останова" и перехода в пультовой режим - запрос возникает при обнаружении низкого уровня на входе <b>nIRQ1</b> . Если уровень не будет снят после обработки запроса, то возникнет новый запрос, так как вход чувствителен к уровню, а не к фронту или срезу. Может быть замаскировано единичными битами <b>PSW10</b> или <b>PSW11</b> (запрещается, если хотя бы один бит установлен)	160002 <sub>8</sub>	177674 <sub>8</sub>	<b>PSW10</b> <b>PSW11</b>
9	Прерывание от BE-таймера, есть только в K1801BM1Г, может быть замаскировано обнулением бита <b>MON</b> в регистре управления таймером. Также прерывание может быть запрещено битами <b>PSW7</b> и <b>PSW10</b> (запрещается, если хотя бы один бит установлен)	000270 <sub>8</sub>	SP-4	<b>PSW10</b> <b>PSW7</b> <b>MON</b>
10	Прерывание по ниспадающему фронту на входе <b>nIRQ2</b> , может быть замаскировано битами <b>PSW7</b> и <b>PSW10</b> (запрещается, если хотя бы один бит установлен)	000100 <sub>8</sub>	SP-4	<b>PSW10</b> <b>PSW7</b>
11	Прерывание по ниспадающему фронту на входе <b>nIRQ3</b> , может быть замаскировано битами <b>PSW7</b> и <b>PSW10</b> (запрещается, если хотя бы один бит установлен)	000270 <sub>8</sub>	SP-4	<b>PSW10</b> <b>PSW7</b>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

	Описание	Вектор	PSW/PC	Маска
- 12	Векторное прерывание от внешних устройств - запрос возникает при обнаружении низкого уровня на входе <b>nVIRQ</b> . Если уровень не будет снят после обработки запроса, то возникнет новый запрос, так как вход чувствителен к уровню, а не к фронту или срезу.	внешний	SP-4	<b>PSW10</b> <b>PSW7</b>
13	Инструкция <b>HALT</b> , программное исключение, единственное программное исключение, вызывающее переход в пультовой режим. Остальные исключения пультового режима вызываются аппаратными причинами.	160002 <sub>8</sub>	177674 <sub>8</sub>	-
14	Инструкция <b>BPT</b> , программное исключение	000014 <sub>8</sub>	SP-4	-
15	Инструкция <b>IOT</b> , программное исключение	000020 <sub>8</sub>	SP-4	-
16	Инструкция <b>EMT</b> , программное исключение	000030 <sub>8</sub>	SP-4	-
17	Инструкция <b>TRAP</b> , программное исключение	000034 <sub>8</sub>	SP-4	-

Если прерывание использует вектор 1600xx<sub>8</sub>, то перед сохранением текущих **PSW** и **PC** осуществляется установка бита 3 по адресу 177716<sub>8</sub> (этот адрес НЕ зависит от физического номера процессора **PA0-PA1**) - таким способом осуществляется переход в "пультовой" режим, и сохранение состояния происходит уже в памяти "пультового" режима. Внутри собственно процессора нет специального "пультового" режима и каких-то выделенных для этого структур, бит управления режимом вынесен во внешние схемы. Подобная реализация делает сложным построение многопроцессорных систем – процессоры разделяют стек пультового режима и бит управления режимом (бит 3 регистра 177716<sub>8</sub>), разделение этих ресурсов может быть возложено на внешние по отношению к процессорам схемы.

При возникновении запроса на прерывание или исключение процессор выполняет следующие операции:

- при необходимости откатывает значение **PC=PC-2** (зависит от младшего бита **MCIR**), если текущая инструкция еще не была обработана, таким образом, сохраняемый **PC** будет указывать на инструкцию-причину исключения
- если необходимо осуществить переход в пультовой режим (код **MCIR=01x<sub>2</sub>**), то на внешней шине выполняется цикл чтение-модификация-запись с установкой бита 3 в регистре 177716<sub>8</sub> (адрес НЕ зависит от физического номера процессора **PA0-PA1**)
- сохраняется текущее значение **PSW** по адресу **-(SP)** или 177676<sub>8</sub> (с использованием внутреннего регистра **R12**, если осуществляется переход в пультовой режим)
- сохраняется текущее значение **PC** по адресу **-(SP)** или 177674<sub>8</sub> (с использованием внутреннего регистра **R12**, если осуществляется переход в пультовой режим)
- если для сохранения использовался регистр **SP**, то он остается модифицированным (уменьшен на 4)
- получает и сохраняет во внутреннем служебном регистре адрес вектора прерывания
- чтение нового значения **PC** по адресу вектор прерывания + 0
- чтение нового значения **PSW** по адресу вектор прерывания + 2
- возобновление выполнения программы
- если на каком-то этапе возникает ошибка шины МПИ (для K1801BM1 возможен только тайм-аут шины, так как исключение при словном обращении по нечетному адресу в данном процессоре отключено), то микропрограммный автомат останавливается, сбрасывается и начинает новый цикл с опроса блока прерывания, при этом он получает код возникшего исключения и переходит к обработке соответствующей ошибки - тайм-аут шины, тайм-аут получения вектора от внешнего устройства, двойной тайм-аут шины



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Описанная выше последовательность действий выполняется как единая процедура за несколько последовательных микропрограммных шагов и может быть прервана только аппаратным сбросом процессора или ошибкой шины МПИ при обработке исключения (то есть, только тайм-аутом шины в случае процессора K1801BM1).

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 8. Описание микропрограммы

В данном разделе приведена таблица всех возможных переходов между микроадресами микропрограммного автомата. В поле MA содержится исходный микроадрес, в поле IR значение регистра инструкций при котором происходит переход, поле MCIR описывает необходимое условие для осуществления перехода, поле NA содержит значение микроадреса, на который будет осуществлен переход при выполнении условий. От поля флагов переходы практически не зависят, немногие реальные случаи даны в описании.

На темном фоне приведены микроадреса, которые приводят к исключению неопределенной инструкции. В состоянии процессора на этих микроадресах не вносятся никаких изменений.

Описание переходов между адресами микроинструкций программного автомата 1801BM1A:

MA	MCIR	Значение IR	NA	Примечание
00	xxx	xxxxxx <sub>8</sub>	01	Начальная точка входа после аппаратного сброса, запускает опрос блока прерываний и переходит в начало основного цикла микропрограммного автомата (адрес 01) MCIR poll
01	000	xxxxxx <sub>8</sub>	01	Постоянный опрос статуса прерываний, используется при начальном старте для ожидания деактивации <b>nACLO</b> , а также инструкцией WAIT, микропрограммный автомат остается на адресе 01 (постоянно переходит сам на себя) и ожидает ненулевого <b>MCIR</b> для выхода из цикла опроса. При выполнении инструкции WAIT установлен флаг <b>WCPU</b> , блок прерываний постоянно записывает значение 000 в <b>MCIR</b> и поэтому содержимое <b>IR</b> не важно и более не декодируется до сброса флага <b>WCPU</b>
	001	000000 <sub>8</sub>	01	Инструкция HALT, записывает код исключения непосредственно в <b>MCIR</b> и перестартовывает исполнение с адреса 01 опрос блока прерываний не запускается, начнется обработка исключения по вектору 160002 <sub>8</sub> в пультовом режиме MCIR = 011 <sub>2</sub> VSEL = 1011 <sub>2</sub> (160002 <sub>8</sub> )
		000003 <sub>8</sub>	01	Инструкция BPT, записывает код исключения непосредственно в <b>MCIR</b> и перестартовывает исполнение с адреса 01, опрос блока прерываний не запускается, начнется обработка исключения по вектору 000014 <sub>8</sub> в пользовательском режиме MCIR = 101 <sub>2</sub> VSEL = 0011 <sub>2</sub> (000014 <sub>8</sub> )

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

MA	MCIR	Значение IR	NA	Примечание
		000004 <sub>8</sub>	01	Инструкция IOT, записывает код исключения непосредственно в <b>MCIR</b> и перестартует исполнение с адреса 01, опрос блока прерываний не запускается, начнется обработка исключения по вектору 000014 <sub>8</sub> в пользовательском режиме  MCIR = 101 <sub>2</sub> VSEL = 0001 <sub>2</sub> (000020 <sub>8</sub> )
		00010x <sub>8</sub>	01	Инструкция JMP dd (безусловный переход с кодом адресации приемника 0), записывает код исключения непосредственно в <b>MCIR</b> и перестартует исполнение с адреса 01, опрос блока прерываний не запускается, начнется обработка исключения по вектору 000004 <sub>8</sub> в пользовательском режиме  MCIR = 101 <sub>2</sub> VSEL = 0100 <sub>2</sub> (000004 <sub>8</sub> )
		004x0x <sub>8</sub>	01	Инструкция JSR r, dd (вызов подпрограммы с кодом адресации приемника 0), записывает код исключения непосредственно в <b>MCIR</b> и перестартует исполнение с адреса 01, опрос блока прерываний не запускается, начнется обработка исключения по вектору 000004 <sub>8</sub> в пользовательском режиме  MCIR = 101 <sub>2</sub> VSEL = 0100 <sub>2</sub> (000004 <sub>8</sub> )
		1040nn <sub>8</sub>	01	Инструкция EMT, записывает код исключения непосредственно в <b>MCIR</b> и перестартует исполнение с адреса 01, опрос блока прерываний не запускается, начнется обработка исключения по вектору 000030 <sub>8</sub> в пользовательском режиме  MCIR = 101 <sub>2</sub> VSEL = 0110 <sub>2</sub> (000030 <sub>8</sub> )
		1044nn <sub>8</sub>	01	Инструкция TRAP, записывает код исключения непосредственно в <b>MCIR</b> и перестартует исполнение с адреса 01, опрос блока прерываний не запускается, начнется обработка исключения по вектору 000034 <sub>8</sub> в пользовательском режиме  MCIR = 101 <sub>2</sub> VSEL = 1100 <sub>2</sub> (000034 <sub>8</sub> )
		077xxx <sub>8</sub>	27	Инструкция SOB  MCIR = 011 <sub>2</sub>
		00001x <sub>8</sub>	2F	Инструкции START, STEP  R12 = 177676 <sub>8</sub> A = R12

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

MA	MCIR	Значение IR	NA	Примечание
		x05x0 <sub>x8</sub> x0600 <sub>x8</sub> x0610 <sub>x8</sub> x0620 <sub>x8</sub> x0630 <sub>x8</sub> 00670 <sub>x8</sub>	31	Одноадресные инструкции CLR(B), COM(B), INC(B), DEC(B), NEG(B), ADC(B), SBC(B), TST(B), ROR(B), ROL(B), ASR(B), ASL(B), SXT с кодом адресации приемника равным 0 (регистр общего назначения). Если в приемнике используется <b>PC</b> , то обновление <b>PC</b> откладывается на момент после фактического выполнения операции, вместо этого временно используется регистр <b>R12</b>  A = PC IR preload start word read start if (dd register == PC) R12 = PC + 2 else PC = PC + 2
		000001 <sub>8</sub>	37	Инструкция WAIT PC = PC + 2
		000005 <sub>8</sub>	37	Инструкция INIT MCIR = 010 <sub>2</sub> , INIT assert
		0064xx <sub>8</sub>	37	Инструкция MARK R12 = PC + IR[5:0]*2
		00024x <sub>8</sub>	37	Инструкции манипуляции флагами CLx, Sex A = PC IR preload start word read start
		000002 <sub>8</sub> 000006 <sub>8</sub>	38	Инструкции RTI, RTT A = SP word read start SP = SP + 2
		0001xx <sub>8</sub>	45	JMP, если метод адресации не 0 MCIR = 100 <sub>2</sub>
		0003xx <sub>8</sub>	45	SWAB, если метод адресации не 0 MCIR = 100 <sub>2</sub>
		x04xxx <sub>8</sub>	45	JSR, если метод адресации не 0 MCIR = 100 <sub>2</sub>
		x05xxx <sub>8</sub>	45	CLR(B), COM(B), INC(B), DEC(B), NEG(B), ADC(B), SBC(B), TST(B), если метод адресации не 0 MCIR = 111 <sub>2</sub> (101 <sub>2</sub> для TST)
		x06xxx <sub>8</sub>	45	ROL(B), ROR(B), ASL(B), ASR(B), SXT, MTPS, MFPS, если метод адресации не 0 MCIR = 111 <sub>2</sub>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

MA	MCIR	Значение IR	NA	Примечание
		xxxxxx <sub>8</sub>	47	Двухадресные инструкции, выполнение арифметических операций XOR, MOV(B), CMP(B), BIT(B), BIC(B), BIS(B), ADD, SUB с кодами адресации приемника и источника (для двухадресных операций) не равными 0 (регистр общего назначения).
		00020x <sub>8</sub>	50	Инструкция RTS MCIR = 011 <sub>2</sub>
		00030x <sub>8</sub>	50	Инструкция SWAB Rd
		10640x <sub>8</sub>	50	Инструкция MTPS Rd
		10670x <sub>8</sub>	50	Инструкция MFPS Rd
		074x0x <sub>8</sub> x10x0x <sub>8</sub> x20x0x <sub>8</sub> x30x0x <sub>8</sub> x40x0x <sub>8</sub> x50x0x <sub>8</sub> x60x0x <sub>8</sub>	58	Выполнение арифметических операций XOR, MOV(B), CMP(B), BIT(B), BIC(B), BIS(B), ADD, SUB с кодами адресации приемника и источника (для двухадресных операций) равными 0 (регистр общего назначения). Если в приемнике или источнике используется <b>PC</b> , то обновление <b>PC</b> откладывается на момент после фактического выполнения операции, вместо этого временно используется регистр <b>R12</b>  A = PC IR preload start word read start if (ss register == PC    dd register == PC) R12 = PC + 2 else PC = PC + 2
		074xxx <sub>8</sub>	5A	Инструкция XOR, если режим адресации приемника != 0 (не регистр общего назначения) MCIR = 011 <sub>2</sub>
		x00xxx <sub>8</sub>	7B	Инструкции условного и безусловного перехода Vxx MCIR = 010 <sub>2</sub>
	001	000007 <sub>8</sub>		Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>
		00002x <sub>8</sub>		
		00003x <sub>8</sub>		
		00004x <sub>8</sub>		
		00005x <sub>8</sub>		
		00006x <sub>8</sub>		
		00007x <sub>8</sub>		
		00021x <sub>8</sub> 00022x <sub>8</sub> 00023x <sub>8</sub>		
		0065xx <sub>8</sub> 0066xx <sub>8</sub>		

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

MA	MCIR	Значение IR	NA	Примечание
		007xxx <sub>8</sub>		
		070xxx <sub>8</sub>		
		071xxx <sub>8</sub>		
		072xxx <sub>8</sub>		
		073xxx <sub>8</sub>		
		075xxx <sub>8</sub>		
		076xxx <sub>8</sub>		
		1065xx <sub>8</sub>		
		1066xx <sub>8</sub>		
		107xxx <sub>8</sub>		
		17xxxx <sub>8</sub>		
	01x	xxxxxx <sub>8</sub>	1D	Переход на процедуру обработки прерывания или исключения в пультовом режиме (core exception) R12 = 177716 <sub>8</sub> (no N <sub>cpu</sub> added) A = R12 word read-modify-write start
	10x	xxxxxx <sub>8</sub>	13	Переход на процедуру обработки прерывания или исключения в пользовательском режиме (user exception) SP = SP - 2 A = SP word write start
	11x	xxxxxx <sub>8</sub>	21	Переход на процедуру начального пуска R12 = 177716 <sub>8</sub> + N <sub>cpu</sub> *20 <sub>8</sub> A = R12 word read start
02	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
03				38:00208100 [x xxx xxx xxx xxx xxx] [00x] 3C:00DF7F00 [x xxx xxx xxx xxx xxx] [xxx]
04				30:00200000 [x xxx xxx xxx 000 xxx] [xxx] 45:00E00000 [x xxx xxx xxx xxx xxx] [xxx]
05 09	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
0A	0xx	xxxxxx <sub>8</sub>	11	Часть общей процедуры извлечения <b>PC</b> и <b>PSW</b> из вектора, используется как ветка для входа прерываний пультового режима MCIR = 0102, WCPU reset
	1xx	xxxxxx <sub>8</sub>	0F	Часть общей процедуры извлечения <b>PC</b> и <b>PSW</b> из вектора, обнуление старшего байта <b>PSW</b> , используется как ветка для прерываний пользовательского режима PSW = PSW & ~177400 <sub>8</sub>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

MA	MCIR	Значение IR	NA	Примечание
0B	xxx	xxxxxxx <sub>8</sub>	3C	Часть общей процедуры извлечения <b>PC</b> и <b>PSW</b> из вектора, запуск чтения <b>PC</b> из вектора в <b>Q</b>  wait qbus or vector read R12 = Q A = Q word read start
0C 0E	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
0F	xxx	xxxxxxx <sub>8</sub>	11	Часть общей процедуры извлечения <b>PC</b> и <b>PSW</b> из вектора  MCIR = 0102, WCPU reset
10				19:00400000 [x xxx xxx x11 xxx xxx] 39:00C00000 [x xxx xxx xxx xxx xxx]
11	xxx	xxxxxxx <sub>8</sub>	01	Завершение исполнения текущей инструкции и подготовка к обработке следующей инструкции – запуск опроса блока прерываний, запуск чтения кода следующей инструкции, продвижение <b>PC</b>  MCIR poll, IR cmd load start, A = PC, PC = PC + 2, word read start
12	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
13	xxx	xxxxxxx <sub>8</sub>	14	Часть общей процедуры входа в исключение или прерывание, сохранение <b>PSW</b> или в стеке <b>SP</b> или по адресу в <b>R12</b>  13: Q = PSW, write data
14	xxx	xxxxxxx <sub>8</sub>	17	Часть общей процедуры входа в исключение или прерывание, декремент указателя стека, <b>R12</b> для пультовых прерываний и <b>SP</b> – для пользовательских, запуск транзакции записи  wait write R12/SP = R12/SP - 2 A = R12/SP word write start
15	xxx	00020x <sub>8</sub>	40	Инструкция RTS, извлечение слова из стека  A = SP, SP = SP + 2, word read start
		нет		40:00400000 [x xxx xxx xxx 00x xxx] 48:00C00000 [x xxx xxx xxx xxx xxx]
16	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
17	xxx	xxxxxxx <sub>8</sub>	1E	Часть общей процедуры входа в исключение или прерывание, выполняется коррекция <b>PC</b> , если есть необходимость  PC = PC - (MCIR[0] ? 0 : 2) Q = PC, write data

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

MA	MCIR	Значение IR	NA	Примечание
18	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
19		xxxxxx <sub>8</sub>	11	plx:01000000 [x xxx xxx xxx xxx] [xxx] [xxxxx] +wait:01000000 [xxxxxxxxxxxxxxx xxx] [xxx] [xxxxx] -init:01000000 [xxxx xxxxxx xxx xxx] [xxx] [xxxxx] +err7:01000000 [xxxxxxxxxxx xxx xxx] [xxx] [xxxxx] 11:01000000 [x xxx xxx xxx xxx] [xxx] [xxxxx]
1A		xxxxxx <sub>8</sub>	41	1000000
1B	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
1C	xxx	xxxxxx <sub>8</sub>	13	Часть процедуры входа в пультовое исключение или прерывание wait write R12 = 177676 <sub>8</sub> A = R12 word write start
1D	xxx	xxxxxx <sub>8</sub>	1C	Часть процедуры входа в пультовое исключение или прерывание wait read R12 = Q   000010 <sub>8</sub> write modified data start
1E	xxx	xxxxxx <sub>8</sub>	0B	Часть общей процедуры получение адреса вектора в <b>Q</b> , получение адреса вектора для чтения новых значений <b>PC</b> и <b>PSW</b>  if nVIRQ wait qbus vector read start else R12 = vector[VSEL] A = R12 Q = R12
1F	xxx	xxxxxx <sub>8</sub>	0A	Часть общей процедуры извлечения <b>PC</b> и <b>PSW</b> из вектора, ожидание чтения <b>PSW</b>  wait read PSW = Q
20	xxx	xxxxxx <sub>8</sub>	11	Часть процедуры начального пуска  PSW = 000340 <sub>8</sub>
21	xxx	xxxxxx <sub>8</sub>	22	Часть процедуры начального пуска  wait read R12 = Q & 177400 <sub>8</sub>
22	xxx	xxxxxx <sub>8</sub>	20	Часть процедуры начального пуска  PC = R12
23				28:00800000 [x xxx xx0 xxx xxx xxx] 38:00800000 [x xxx xx1 xxx xxx xxx]



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

MA	MCIR	Значение IR	NA	Примечание
24 25	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
26				30:00950000 [x xxx xxx xxx xxx] [xxx] 32:002B0000 [0 xxx x0x xxx xxx xxx] [xxx] 38:00400000 [0 xxx x1x xxx xxx xxx] [xxx]
27	xxx	xx{xx0 <sub>2</sub> }xxx <sub>8</sub>	26	26:00800000 [x xxx xx0 xxx xxx xxx] Rr = Rr & 177777 <sub>8</sub>
	xxx	xx{xx1 <sub>2</sub> }xxx <sub>8</sub>	37	Обработка инструкции SOB, декремент регистра Rr = Rr - 1
28 29	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
2A	xxx	xxxxxxx <sub>8</sub>	11	Инструкции START wait read, Q &= ~10 <sub>8</sub> , modified write
	xxx	xxxxxxx <sub>8</sub>	7A	Инструкции STEP wait read, Q &= ~10 <sub>8</sub> , modified write
				11:00400000 [x xxx xxx xxx xx1 0xx] [xxx] [xxxxx] 30:00200000 [x xxx xxx xxx xx0 0xx] [xxx] [1xxxx] 31:00200000 [x xxx xxx xxx xx0 0xx] [xxx] [0xxxx] 38:00400000 [x xxx xxx 1xx xxx 1xx] [xxx] [xxxxx] 3A:00200000 [x xxx xxx 0xx xx0 1xx] [xxx] [xxxxx] 7A:00200000 [x xxx xxx 0xx xx1 1xx] [xxx] [xxxxx]
2B	xxx	xxxxxxx <sub>8</sub>	2A	Инструкции START, STEP R12 = 1777168, A = R12, start rmw
				2A:00200000 [x xxx xx0 0xx xx1 xxx] [xxx] [xxxxx] 38:00800000 [x xxx xxx 1xx xxx xxx] [xxx] [xxxxx] 3A:00600000 [x xxx xxx 0xx xxx xxx] [xxx] [xxxxx]
2C	xxx	xxxxxxx <sub>8</sub>	2B	Инструкции START, STEP wait read, PC = Q
2D	xxx	xxxxxxx <sub>8</sub>	2C	Инструкции START, STEP R12 = R12 - 2, A = R12, word read
2E	xxx	00001x <sub>8</sub>	2D	Инструкции START, STEP wait read, PC = Q
		xxxxxxx <sub>8</sub>		2D:00800000 [x xxx xxx xxx xx1 xxx] [xxx] [xxxxx] 3C:00400000 [x xxx xxx xxx xx0 xxx] [xxx] [1xxxx] 3D:00400000 [x xxx xxx xxx xx0 xxx] [xxx] [0xxxx]
2F	xxx	000002 <sub>8</sub> 000006 <sub>8</sub>	36	Инструкции RTI, RTT, если T в PSW единичный PSW = PSW & ~177400 <sub>8</sub>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

MA	MCIR	Значение IR	NA	Примечание
	xxx	000002 <sub>8</sub> 000006 <sub>8</sub>	37	Инструкции RTI, RTT, если <b>T</b> в <b>PSW</b> нулевой PSW = PSW & ~177400 <sub>8</sub>
	xxx	00001x <sub>8</sub>	2E	Инструкции START, STEP R12 = R12 - 2, A = R12, word read
				2E:00400000 [x xxx xx0 xxx xx1 xxx] [xxx] [xxxxx] 36:00400000 [x xxx xxx xxx xx0 xxx] [xxx] [1xxxx] 37:00400000 [x xxx xxx xxx xx0 xxx] [xxx] [0xxxx] 3E:00400000 [x xxx xx1 xxx xx1 xxx] [xxx] [xxxxx]
30		xxxxxx <sub>8</sub>	59	1000000
31		xxxxxx <sub>8</sub>	49	1000000
32	xxx	0001dd <sub>8</sub>	11	Инструкция JMP PC = R12
				11:00FC0000 [x xxx xxx xxx xxx xxx] 30:00040000 [x xxx 0xx x00 xxx 1x1]
33	xxx	00020x <sub>8</sub>	15	Инструкция RTS PC = R12
		нет		11:00480000 [0 xxx xxx xxx xxx xxx] 15:00180000 [0 xxx x00 xxx xxx xxx] 30:00580000 [x xxx xxx xxx xxx xxx] 34:00480000 [x xxx xx0 xxx xxx xxx]
34 35	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
36				uop:00008000 [0 xxx 000 00x 0xx xxx] [xxx] [xxxxx] plx:0001A000 [0 xxx 000 0xx 0xx xxx] [xxx] [xxxxx] +wait:0000C000 [0xxx000011 0xx xxx] [xxx] [xxxxx] -init:00004000 [0xxx 000 00x 0xx xx1] [1xx] [xx1xx] +err7:0000C000 [0xxx0000110xx xxx] [xxx] [xxxxx] 01:00068000 [0 xxx 000 xxx xxx xxx] [xxx] [xxxxx] 07:00020000 [0 xxx 000 xxx 0xx xxx] [xxx] [xxxxx] 11:006A0000 [x xxx xxx xxx xxx xxx] [xxx] [xxxxx] 27:000F8000 [0 xxx 100 xxx xxx xxx] [xxx] [xxxxx] 28:00004000 [0 xxx 000 00x 1xx 1x0] [xxx] [1xxxx] 2A:00008000 [0 xxx 100 00x xxx 1x0] [xxx] [1xxxx] 30:0002C000 [x xxx 0xx x00 xxx 1x1] [xxx] [xx0xx] 31:00340000 [x xxx xxx xxx xxx xxx] [xxx] [xxxxx] 36:00004000 [0 xxx 000 x00 0xx 1x1] [xxx] [xx0xx] 37:00008000 [0 xxx 000 x0x 0xx xx1] [0xx] [xx1xx] 38:00080000 [x xxx xxx x0x xxx 1xx] [xxx] [xxxxx] 39:003D0000 [0 xxx x1x xxx xxx xxx] [xxx] [xxxxx] 47:00008000 [0 xxx 000 00x 0xx xxx] [xxx] [xxxxx] 6A:00004000 [0 xxx 000 00x 0xx 1x0] [xxx] [1xxxx]

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

MA	MCIR	Значение IR	NA	Примечание
	xxx	000002 <sub>8</sub>	01	Обработка инструкции RTI при установленном <b>T</b> MCIR = 101 <sub>2</sub> , VSEL = 0011 <sub>2</sub>
	1xx	000005 <sub>8</sub>	11	Обработка инструкции INIT, завершение цикла if (FR Z) INIT reset GOTO 11 else R12 = R12 - 2
	xxx	000005 <sub>8</sub>	36	Обработка инструкции INIT if (FR Z) INIT reset GOTO 11 else R12 = R12 - 2
	0xx	000005 <sub>8</sub>	37	Обработка инструкции INIT if (FR Z) INIT reset MCIR = 110 <sub>2</sub> GOTO 37 else R12 = R12 - 2
	xxx	000006 <sub>8</sub>	6A	Обработка инструкции RTT при установленном <b>T</b> A = PC, IR cmd preload start, PC = PC + 2
	xxx	00024x <sub>8</sub>	01	Инструкции манипулирования флагами CLx, SEx 36: PSW = PSW & ~R12, MCIR poll

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

MA	MCIR	Значение IR	NA	Примечание
37				uop:00088000 [x xxx xx0 00x xxx xxx] [xxx] [xxxxxx] plx:000D0000 [x xxx xx0 0xx xxx xxx] [xxx] [xxxxxx] +wait:000D0000 [xxxxxx00xxxxx xxx] [xxx] [xxxxxx] -wait:000A0000 [1xxxxx00xx xxx xxx] [xxx] [xxxxxx] +err7:000C0000 [xxxxxx00xxxxx xxx] [xxx] [xxxxxx] 01:00044000 [0 xxx 000 x0x xxx xxx] [xxx] [xxxxxx] 07:00018000 [0 xxx 000 10x xxx xxx] [xxx] [xxxxxx] 11:002A8000 [0 xxx xxx xxx xxx xxx] [xxx] [xxxxxx] 27:00100000 [0 xxx 100 xxx xxx xxx] [xxx] [xxxxxx] 2A:00002000 [0 xxx 000 100 0xx 1x0] [xxx] [1xxxxx] 30:00614000 [x xxx xxx xxx xxx xxx] [xxx] [xxxxxx] 31:00100000 [0 xxx 101 xxx xxx xxx] [xxx] [xxxxxx] 36:00420000 [x xxx xx0 xxx xxx xxx] [xxx] [xxxxxx] 38:00020000 [x xxx 0x1 xx0 0xx 1x0] [xxx] [1xxxxx] 3A:00018000 [x xxx 0x0 xx0 0xx 1x0] [xxx] [1xxxxx] 47:00018000 [0 xxx 000 00x xxx xxx] [xxx] [xxxxxx] 6A:00002000 [0 xxx 000 000 0xx 1x0] [xxx] [1xxxxx] 76:00070000 [1 xxx xx0 00x xxx xxx] [xxx] [xxxxxx] 7A:00004000 [1 xxx 0x0 000 0xx 1x0] [xxx] [1xxxxx]
	xxx	000001 <sub>8</sub>	01	Обработка инструкции WAIT MCIR = 000 <sub>2</sub> , set WCPU
	xxx	000002 <sub>8</sub>	01	Обработка инструкции RTI (при сброшенном T) MCIR poll, A = PC, IR cmd start, PC = PC + 2
	xxx	000005 <sub>8</sub>	36	Обработка инструкции INIT R12 = 000340 <sub>8</sub>
	xxx	000006 <sub>8</sub>	01	Обработка инструкции RTT (при сброшенном T) A = PC, IR cmd load start, PC = PC + 2
	xxx	00024x <sub>8</sub>	36	Инструкции манипулирования флагами CLx, SEx 37: R12 = IR[3:0]
	xxx	077xxx <sub>8</sub>	11	Обработка условного перехода инструкции SOB if (FR Z) R12 = PC - 2*IR[5:0]; else PC = PC - 2*IR[5:0];
38	xxx	xxxxxx <sub>8</sub>	39	Инструкции RTI, RTT Запись прочитанного значения в <b>PC</b> или <b>R12</b> wait read if (IR[2:0] != 7) PC = Q else R12 = Q

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

MA	MCIR	Значение IR	NA	Примечание
39	xxx	xxxxxx <sub>8</sub>	6B	Инструкции RTI, RTT Запуск чтения слова из стека ( <b>SP</b> )  A = SP word read start SP = SP + 2
3A 3B	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
3C	xxx	xxxxxx <sub>8</sub>	3D	Часть общей процедуры извлечения <b>PC</b> и <b>PSW</b> из вектора, ожидание чтения <b>Q</b> из вектора и запись в <b>PC</b>  wait read PC = Q
3D	xxx	xxxxxx <sub>8</sub>	1F	Часть общей процедуры извлечения <b>PC</b> и <b>PSW</b> из вектора, запуск чтения <b>PSW</b> из вектора по адресу в <b>R12</b>  R12 = R12 + 2 A = R12 word read start
3E 3F	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
40	xxx	xxxxxx <sub>8</sub>	11	Инструкция RTS  wait read, Rd = Q
41				10:00000900 [x xxx xx0 xxx 000 xxx] [001] [xxxxx] 11:003EBFA0 [x xxx xxx xxx xxx xxx] [x0x] [xxxxx] 18:002FC400 [x xxx xxx xxx xxx xxx] [xxx] [xxxxx] 19:004F8500 [x xxx xxx xxx xxx xxx] [xxx] [xxxxx] 31:00200620 [x xxx xxx xxx xxx xxx] [xxx] [xxxxx] 38:00107C40 [x xxx xxx xxx xxx xxx] [xxx] [xxxxx] 39:00116C00 [x xxx xxx xxx xxx xxx] [xxx] [xxxxx]
42				04:00800000 [x xxx xx0 xxx xxx xxx] 4E:00800000 [x xxx xx1 xxx xxx xxx]
43 44	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
45	xxx	xxxx0x <sub>8</sub>	47	Одноадресные инструкции с методом адресации 0 вызывают исключения, на данную точку нет перехода для таких инструкций, код вставлен для перестраховки
	xxx	xxxx1x <sub>8</sub>	48	Одноадресные инструкции с методом адресации 1
	xxx	xxxxxx <sub>8</sub>	4C	Одноадресные инструкции с методами адресации 6 и 7
	xxx	xxxxxx <sub>8</sub>	4D	Одноадресные инструкции с методами адресации 3 и 5
	xxx	xxxxxx <sub>8</sub>	55	Одноадресные инструкции с методами адресации 2 и 4

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

MA	MCIR	Значение IR	NA	Примечание
46				04:00800000 [x xxx xx0 xxx xxx xxx] 4E:00800000 [x xxx xx1 xxx xxx xxx]
47				04:00600000 [x xxx xxx xxx xxx xxx] 46:00400000 [x xxx 11x xxx xxx xxx] 4E:00400000 [x xxx xx1 xxx xxx xxx] 5A:00200000 [x xxx 000 xxx xxx xxx]
48				Используется двухадресными инструкциями для финальной операции, MCIR при этом 0xx <sub>2</sub> , поэтому никакие флаги в дополнительном векторе не устанавливаются plx:0036C000 [x x0x xxx xxx xxx xxx] [1xx] [xxxxxx] +wait:0036C000 [xx0xxxxxxxxxxx xxx] [1xx] [xxxxxx] -wait:00019800 [xx00xxxxxxxx xxx xxx] [1xx] [xxxxxx] +init:0036C000 [xx0xxxxxxxx xxx xxx] [1xx] [xxxxxx] -init:00290000 [xx0x xxx xxx xxx xxx] [1xx] [xxxxxx] +stop:00098000 [xx0xx1xxxx xxx xxx] [1xx] [xxxxxx] +err7:0036C000 [xx0xxxxxxxxxxx xxx] [1xx] [xxxxxx] 10:00016000 [x 000 0xx xxx xxx xxx] [100] [xxxxxx] 11:00A96000 [x xxx xxx xxx xxx xxx] [xxx] [xxxxxx] 12:00002000 [x 000 010 1x0 xxx xxx] [100] [xxxxxx] 17:0030A000 [x x0x xxx xxx xxx xxx] [1xx] [xxxxxx] 19:00200000 [x 00x xxx xxx xxx xxx] [0xx] [xxxxxx] 31:00004000 [1 000 x11 1x0 xxx xxx] [1x1] [xxxxxx] 32:00008000 [x 000 000 xxx xxx xxx] [100] [xxxxxx] 37:0003C000 [x 000 xx0 xxx xxx xxx] [1xx] [xxxxxx]
49				plx:00550000 [x x0x xxx xxx xxx xxx] [xxx] [xxxxxx] +wait:00550000 [xx0xxxxxxxxxxx xxx] [xxx] [xxxxxx] -wait:00010000 [x000x100x1xxx xxx] [1xx] [xxxxxx] +stop:00090000 [xx0xx1x0xx xxx xxx] [1xx] [xxxxxx] 01:00976000 [x xxx xxx xxx xxx xxx] [xxx] [xxxxxx] 11:0015A000 [x xxx xxx xxx xxx 111] [xxx] [xxxxxx] 47:0048A000 [x x0x xxx xxx xxx xxx] [xxx] [xxxxxx] 57:000A6000 [x x0x xxx xxx xxx 111] [xxx] [xxxxxx]
4A 4B	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
4C				48:00800000 [x xxx xxx xxx xx0 xxx] 4D:00800000 [x xxx xxx xxx xx1 xxx]
4D	xxx	xxxxxx <sub>8</sub>	48	1000000
4E	xxx	xxxxxx <sub>8</sub>	04	1000000
4F				
50	xxx	00020x <sub>8</sub>	33	Инструкция RTS R12 = Rd & 177777 <sub>8</sub>
	xxx	00030x <sub>8</sub>	51	Инструкция SWAB Rd

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

MA	MCIR	Значение IR	NA	Примечание
	xxx	10640 <sub>x8</sub>	37	Инструкция MTPS Rd
	xxx	10670 <sub>x8</sub>	51	Инструкция MFPS Rd
				Установка +wait и –wait не используются для обрабатываемых инструкций на данном микроадресе plx:00200000 [x xxx xxx 011 xxx xxx] [xxx] [xxxxxx] +wait:00200000 [x xxxxxx011xxxxxx] [xxx] [xxxxxx] -wait:00200000 [xxxxxxx 011 xxx xxx] [xxx] [xxxxxx] 31:00400000 [x xxx xxx x01 xxx xxx] [xxx] [xxxxxx] 33:00400000 [x xxx xxx x10 xxx xxx] [xxx] [xxxxxx] 37:00400000 [x xxx xxx x00 xxx xxx] [xxx] [xxxxxx] 51:00400000 [x xxx xxx x11 xxx xxx] [xxx] [xxxxxx]
51				01:00E00000 [x xxx xxx xxx xxx xxx] 11:00200000 [x xxx xxx xxx xxx 111]
52 54	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
55				uor:00400000 [x xxx xxx xxx 00x xxx] 47:00400000 [x xxx xxx xxx 00x xxx] 48:00C00000 [x xxx xxx xxx xxx xxx]
56 57	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
58				plx:00300000 [x x0x xxx xxx xxx xxx] [xxx] [xxxxxx] +wait:00300000 [x x0x xxx xxx xxx xxx] [xxx] [xxxxxx] -wait:00180000 [xx0xxxxx1x xxx xxx] [xxx] [xxxxxx] +stop:00180000 [xx0xxxxx1x xxx xxx] [xxx] [xxxxxx] +err3:00180000 [xx0xxxxx1x xxx xxx] [xxx] [xxxxxx] 01:0086C000 [x xxx xxx xxx xxx xxx] [xxx] [xxxxxx] 11:00294000 [x xxx xxx xxx xxx xxx] [xxx] [xxxxxx] 47:0024C000 [x x0x xxx xxx xxx xxx] [xxx] [xxxxxx] 51:00200000 [1 00x xxx xxx xxx xxx] [xxx] [xxxxxx] 57:000B4000 [x x0x xxx xxx xxx xxx] [xxx] [xxxxxx]
59				plx:00300000 [x x0x xxx xxx xxx xxx] [xxx] [xxxxxx] +wait:00300000 [xx0xxxxxxxxxxx xxx] [xxx] [xxxxxx] -wait:00300000 [x x0x xxxxxxxxxxxxxx] [xxx] [xxxxxx] +stop:00300000 [xx0xxxxxxxx xxx xxx] [xxx] [xxxxxx] +err3:00300000 [xx0xxxxxxxx xxx xxx] [xxx] [xxxxxx] 01:009A0000 [x xxx xxx xxx xxx xxx] [xxx] [xxxxxx] 11:00160000 [x xxx xxx xxx xxx 111] [xxx] [xxxxxx] 47:002A0000 [x x0x xxx xxx xxx xxx] [xxx] [xxxxxx] 51:00200000 [1 00x xxx xxx xxx xxx] [xxx] [xxxxxx] 57:00060000 [x x0x xxx xxx xxx 111] [xxx] [xxxxxx]
5A	xxx	xxxxxx <sub>8</sub>	45	1000000
5B 69	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

MA	MCIR	Значение IR	NA	Примечание
6A	xxx	000006 <sub>8</sub>	01	Инструкция RTT при установленном флаге <b>T</b> ожидает загрузки кода следующей инструкции
	xxx	000014 <sub>8</sub>	01	Инструкция STEP ожидает загрузки кода следующей инструкции
				01:00800000 [x xxx xxx 0xx xxx xxx] 11:00800000 [x xxx xxx 1xx xxx xxx]
6B	xxx	000002 <sub>8</sub> 000006 <sub>8</sub>	2F	Инструкции RTI, RTT PSW = Q
	xxx			11:00800000 [x xxx xxx 1xx xxx xxx] 2F:00800000 [x xxx xxx 0xx xxx xxx]
6C 79	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
7A	xxx	00001x <sub>8</sub>	6A	Инструкция STEP, запуск чтения следующей инструкции A = PC, IR cmd preload start, PC = PC + 2
				28:00800000 [x xxx xxx xxx xx0 xxx] 6A:00800000 [x xxx xxx xxx xx1 xxx]
7B	xxx	xxxxxx <sub>8</sub>	11	Инструкции условного и безусловного перехода Vxx. При совпадении флагов с условиями, которые требуются в инструкции, происходит добавление смещения к <b>PC</b>  if (cond) PC = PC + sign extended(IR[7:0]*2) else R12 = PC + sign extended(IR[7:0]*2)
7C 7D	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			
7E	xxx	xxxxxx <sub>8</sub>	6A	1000000
7F	Неопределенная инструкция, сброс автомата, исключение по вектору 000010 <sub>8</sub>			



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 8.1 Цикл микропрограммного автомата

Циклом микропрограммного автомата называется набор тактов частоты **CLC**, при котором в регистре **МА** сохраняется некоторое фиксированное значение микроадреса. Цикл микропрограммного автомата длится минимум два такта **CLC**, но может быть удлинён на некоторое количество тактов по следующим причинам:

- ожидание готовности результата АЛУ
- ожидание завершения чтения данных на шине МПИ, когда данные нужны в качестве аргумента для операции АЛУ, данные будут занесены в регистр **Q** и использованы в операции
- ожидания завершения чтения кода инструкции на шине МПИ и записи его в регистр инструкций **IR**
- ожидание завершения опроса блока прерываний
- ожидание освобождения регистра адреса **A** для инициирования новой транзакции на шине МПИ в конвейере

По завершению цикла в регистр **МА** записывается новое значение микроадреса, что означает переход к новому циклу микропрограммного автомата. Иногда значение нового микроадреса может совпадать со старым, что приводит к повторению цикла, вероятно с новыми параметрами. Например, инструкция **BPT** на микроадресе **01** записывает новое значение в регистр **MCIR** и повторяет цикл автомата по тому же микроадресу **01**, но уже с новым параметром.

После аппаратного сброса микропрограммный автомат начинает исполнение с микроадреса **00**, на котором просто происходит запуск опроса блока прерываний и следует безусловный переход на основную точку входа микропрограммы на микроадресе **01**.

Микроадрес **01** является основной точкой входа в цикл обработки процессорных инструкций, при выполнении этого микроадреса обычно уже выполнен опрос блока прерываний и регистр **MCIR** содержит актуальное состояние запросов на прерывание или код исключения. Также предполагается, что в регистр кода инструкции **IR** уже загружено слово новой инструкции, регистр **PC** указывает на следующее слово за уже загруженным в **IR** (**PC** содержит значение “+2” для обрабатываемой инструкции). На микроадресе **01** происходит основное ветвление микропрограммы – начинается обработка прерывания или исключения, или декодирование и исполнение загруженного кода процессорной инструкции.

Особые ситуации возникают при исключениях тайм-аута транзакции на шине МПИ и неопределённой инструкции. При этих исключениях блок прерываний фиксирует соответствующие флаги, а микропрограммный автомат принудительно переводится на микроадрес **01** и также принудительно запускается опрос блока прерываний, гарантируя, что в регистр **MCIR** будет корректно записана причина исключения. Схемами процессора также предполагалось что будет возникать подобное исключение при словном обращении по нечетному адресу, но данная возможность в процессоре **1801BM1** отключена.

Последовательность обработки большинства инструкций завершается на адресе **11** – в этом месте микропрограммный автомат ожидает готовности шины МПИ к запуску транзакции чтения кода следующей процессорной инструкции, записывает значение **PC** в регистр адреса **A**, запуская собственно транзакцию, затем **PC** увеличивается на 2. Также выдается запрос на проверку статуса блока прерываний и происходит переход на адрес **01**, обеспечивая начало обработки следующей инструкции.

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 8.2 Процедура начального старта

Аппаратный сброс микропроцессора 1801BM1 осуществляется низким уровнем на входе **nDCLO**. Микропрограммный автомат при этом сбрасывается и при снятии низкого уровня со входа **nDCLO** начинает исполнение с микроадреса 00. На этом микроадресе осуществляется запуск опроса блока прерываний и происходит переход на микроадрес 01. Блок прерываний будет возвращать нулевое значение **MCIR** до детектирования нарастающего фронта на входе **nACLO**. Основная микропрограмма на микроадресе 01 при нулевом значении **MCIR** перезапускает опрос блока прерываний и повторяет исполнение микроадреса 01. При обнаружении нарастающего фронта на входе **nACLO** (первого после аппаратного сброса по **nDCLO**) блок прерываний вернет значение **MCIR** равное  $111_2$  и микропрограмма по адресу 01 начнет исполнение последовательности начального пуска.

```

00: MCIR poll
01: while (MCIR == 0002) MCIR poll
01: if (MCIR == 112)
    R12 = 1777168 + Ncpu*208
    A = R12
    word read start
21: wait read
    R12 = Q & 1774008
22: PC = R12
20: PSW = 0003408
11: MCIR poll, IR cmd load start,
    A = PC, PC = PC + 2, word read start
01: main loop

```

В ходе выполнения последовательности начального пуска происходит чтение внешнего регистра  $177716_8 + N_{cpu} * 20_8$ , который в старшем байте должен содержать значение, записываемое в старший байт программного счетчика **PC**. Фактический адрес регистра вектора начального пуска зависит от номера процессора в системе (определяется значением на входах **nPA0** и **nPA1**). Для определения начального значения **PC** используется только старший байт содержимого регистра начального пуска. Далее слову состояния процессора **PSW** присваивается фиксированное значение  $340_8$ , затем запускается выборка слова инструкции по адресу в **PC**, **PC** автоинкрементируется, запускается опрос блока прерываний и начинается исполнение основного микропрограммного цикла обработки процессорных инструкций

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

### 8.3 Процедура входа в пультовые исключения и прерывания

Микрокод процедуры входа в пультовые исключения и прерывания имеет вид:

```

01: R12 = 1777168 (no Ncpu*208 added)
    A = R12
    word read-modify-write start
1D: wait read
    R12 = Q | 0000108
    write modified data start
1C: wait write
    R12 = 1776768
    A = R12
    word write start
13: Q = PSW, write data
14: wait write
    R12 = R12 - 2
    A = R12
    word write start
17: PC = PC - ((MCIR == 0112) ? 0 : 2)
    Q = PC, write data
1E: R12 = vector[VSEL]
    A = R12
    Q = R12
0B: wait write
    R12 = Q
    A = Q
    word read start
3C: wait read
    PC = Q
3D: R12 = R12 + 2
    A = R12
    word read start
1F: wait read
    PSW = Q
0A: MCIR = 0102, WCPU reset
11: MCIR poll, IR cmd load start,
    A = PC, PC = PC + 2, word read start
01: main loop

```

Исключения и прерывания пультового режима выполняются следующим образом:

- в начале обработки устанавливают бит 3 в регистре 177716<sub>8</sub>
- сохраняют **PSW** по адресу 177676<sub>8</sub>
- для исключений с кодом **MCIR**=010<sub>2</sub> корректируют **PC** вычитая 2
- сохраняют **PC** по адресу 177674<sub>8</sub>
- читают новое значение **PC** по адресу вектор+0
- читают новое значение **PSW** по адресу вектор+2

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Следующие прерывания и исключения будут обрабатываться процессором в пультовом режиме (с установкой бита 3 в регистре 177716<sub>8</sub>) по векторам:

Инструкция <b>HALT</b>	160002 <sub>8</sub>	<b>PC</b>
Низкий уровень на входе <b>nIRQ1</b>	160002 <sub>8</sub>	<b>PC-2</b>
Тайм-аут шины МПИ, <b>PSW10=1</b> и <b>PSW11=0</b>	160002 <sub>8</sub>	<b>PC</b>
Тайм-аут шины МПИ, 1801BM1A, <b>PSW10=x</b> и <b>PSW11=1</b>	160002 <sub>8</sub>	<b>PC-2</b>
Тайм-аут шины МПИ, 1801BM1Г, <b>PSW10=x</b> и <b>PSW11=1</b>	160002 <sub>8</sub>	<b>PC</b>
Двойной тайм-аут шины МПИ (повторный при попытке обработать первое исключение по зависанию)	160006 <sub>8</sub>	<b>PC</b>
Тайм-аут шины МПИ при получении вектора прерывания (в цикле <b>nIAKO</b> )	160012 <sub>8</sub>	<b>PC</b>

В крайнем правом столбце таблицы указано сохраняемое значение в ячейке 177674<sub>8</sub>, при входе в обработчик исключения по инструкции HALT там будет сохранен указатель на следующую за HALT инструкцию, при возникновении исключения по низкому уровню на входе будет сохранен указатель на первую еще не обработанную инструкцию, на которой возникло исключение. Исключения по тайм-ауту шины обычно (для 1801BM1Г всегда) сохраняют указатели на +2 к базовому адресу инструкции, являющейся причиной исключения.

Следует отметить, что прерывания и исключения пультового режима никак не используют стек пользовательского режима и регистр указателя на него – **SP (R6)** остается неизменным.

Также имеется различие между процессорами 1801BM1A и 1801BM1Г при обработке исключения зависания шины, если установлен бит **PSW11**– сохраняются различные значения **PC**.

Возврат из обработчиков исключений и прерываний пультового режима удобно осуществлять специальными инструкциями процессора 1801BM1, которые автоматически извлекают сохраненные значения **PC** и **PSW** по адресу 177674<sub>8</sub> и сбрасывают бит 3 в регистре 177716<sub>8</sub>. Инструкция START является аналогом стандартной инструкции RTI, инструкция STEP является аналогом стандартной инструкции RTT. Отличие от стандартных инструкций заключается в использовании стека пультового режима и последующем сбросе бита 3 регистра 177716<sub>8</sub>.

Собственно в процессоре 1801BM1 нет никаких аппаратных средств или признаков "пультового" режима - процессор всегда работает одинаково. Реализация "пультового" режима заключается в микропрограммном управлении битом 3 во внешнем (по отношению к микросхеме процессора) регистре, расположенном по адресу 177716<sub>8</sub>. При входе в "пультовый" режим по прерыванию или исключению процессор выполняет операцию чтение-модификация-запись по адресу 177716<sub>8</sub>, устанавливая бит 3. При возврате из "пультового" режима инструкциями STEP и START происходит аналогичное обращение по тому же адресу со сбросом бита 3. Особенности реализации данного бита могут различаться, например, в системах, построенных на основе 1801ВП1-030, данный бит управляет отображением ПЗУ в области адресов 160000<sub>8</sub>-173777<sub>8</sub>, при входе в "пультовый" режим бит 3 будет установлен и по данным адресам будет гарантировано отображено системное ПЗУ, в котором можно разместить гарантированно доступный код обработчика пультового исключения или прерывания.

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

#### 8.4 Процедура входа в пользовательские исключения и прерывания

Микрокод процедуры входа в пользовательские исключения и прерывания имеет вид:

```

01: SP = SP - 2
    A = SP
    word write start
13: Q = PSW, write data
14: wait write
    SP = SP - 2
    A = SP
    word write start
17: PC = PC - ((MCIR == 1012) ? 0 : 2)
    Q = PC, write data
1E: if nVIRQ
    wait write
    vector read start
    else
    R12 = vector[VSEL]
    A = R12
    Q = R12
0B: wait write or vector read
    R12 = Q
    A = Q
    word read start
3C: wait read
    PC = Q
3D: R12 = R12 + 2
    A = R12
    word read start
1F: wait read
    PSW = Q
0A: PSW = PSW & ~1774008
0F: MCIR = 0102, WCPU reset
11: MCIR poll, IR cmd load start,
    A = PC, PC = PC + 2, word read start
01: main loop

```

Исключения и прерывания пользовательского режима выполняются следующим образом:

- **SP = SP – 2**, сохраняют **PSW** по адресу **SP**
- для исключений с кодом **MCIR=100<sub>2</sub>** корректируют **PC** вычитая 2
- **SP = SP – 2**, сохраняют **PC** по адресу **SP**
- читают новое значение **PC** по адресу вектор+0
- читают новое значение **PSW** по адресу вектор+2
- обнуляют старший байт **PSW**

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Следующие прерывания и исключения будут обрабатываться процессором в пользовательском режиме (без модификации бита 3 в регистре 177716<sub>8</sub> и использованием стека SP) по векторам:

тайм-аут внешней шины МПИ, если <b>PSW10=0, PSW11=0</b>	000004 <sub>8</sub>	PC
Недопустимый код инструкции	000010 <sub>8</sub>	PC
<b>T</b> -бит ловушки пошаговой отладки	000014 <sub>8</sub>	PC-2
Пропадание питания, ниспадающий фронт на входе <b>nACLO</b> .	000024 <sub>8</sub>	PC-2
Прерывание от ВЕ-таймера	000270 <sub>8</sub>	PC-2
Прерывание по срезу на входе <b>nIRQ2</b>	000100 <sub>8</sub>	PC-2
Прерывание по срезу на входе <b>nIRQ3</b>	000270 <sub>8</sub>	PC-2
Векторное прерывание по низкому уровню на входе <b>nVIRQ</b>	xxxxxx <sub>8</sub>	PC-2
Инструкция BPT, программное исключение	000014 <sub>8</sub>	PC
Инструкция IOT, программное исключение	000020 <sub>8</sub>	PC
Инструкция EMT, программное исключение	000030 <sub>8</sub>	PC
Инструкция TRAP, программное исключение	000034 <sub>8</sub>	PC

Последний столбец таблицы отображает сохраняемое в стеке значение **PC**. Асинхронные прерывания, вызываемые внешними сигналами, а также отладочное исключение по **T**-биту сохраняют в стеке указатель на первую невыполненную инструкцию, в то время как синхронные исключения сохраняют указатель на следующую инструкцию (или на базу текущей инструкции + 2).

При входе в процедуру обработки прерывания или исключения пользовательского режима также обнуляется старший байт слова состояния процессора **PSW**, значения, прочитанные из вектора, игнорируются.

Для векторного прерывания по низкому уровню на входе **nVIRQ**, значение вектора читается в специальной транзакции подтверждения векторного прерывания на шине МПИ.

Возврат из процедур обработки прерываний и исключений пользовательского режима следует осуществлять при помощи стандартных инструкций RTI и RTT.

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 8.5 Реализация инструкции умножения в процессоре 1801BM1Г

Версия микропрограммы процессора 1801BM1Г дополнительно реализует инструкцию целочисленного умножения.

Инструкция MUL умножает регистровый операнд приемника на стандартный операнд источника, старшие 16 бит произведения сохраняются в регистре приемника, младшие 16 бит произведения сохраняются в регистре с номером регистр приемника плюс 1, если регистровый номер приемника четный, иначе старшие биты просто отбрасываются (просто затираются младшими битами). Код инструкции MUL – 070rss<sub>8</sub>. При попытке исполнения инструкции MUL на процессоре 1801BM1A происходит исключение по вектору 10<sub>8</sub>. Умножение предполагает что числа знаковые, в дополнительном коде.

```

01: MCIR = 1012
if (ss address mode == 0)
    7A: R11 = Rr
    40: R12 = Rs
else
    switch(ss address mode)
        case 0: Rs
            45: A = R12 = Rs, word read start
            58: stop microcode (error)
        case 1: @Rs
            45: A = Rs, R12 = Rs, word read start
        case 2: (Rd)+
            45: A = Rs, Rs = Rs + 2, word read start
            55: R12 = Rs - 2
        case 3: @(Rs)+
            45: A = Rs, word read start, Rs = Rs + 2
            4C: wait read, R12 = Q,
                A = Q, word read start
        case 4: -(Rs)
            45: Rs = Rs - 2, A = Rs, word read start
            55: R12 = Rs
        case 5: @-(Rs)
            45: Rs = Rs - 2, A = Rs, word read start
            4C: wait read, R12 = Q
                A = Q, word read start
        case 6: E(Rs)
            45: A = PC, word read start, PC = PC + 2
            44: wait read, R12 = Rs + Q,
                A = R12, word read start
        case 7: @E(Rs)
            45: A = PC, word read start, PC = PC + 2
            44: wait read, R12 = Rs + Q, A = R12
                word read start
            4C: wait read, R12 = Q,
                A = Q, word read start
    48: wait read, R12 = Q
//
// Аргументы получены в R11 и R12, можно выполнять умножение
//

```

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

```

// R11 - первый аргумент, R12 - второй
// R8 - старшие 16 бит произведения, R9 - младшие
32: R8 = 0
25: R10 = R11
6E: R9 = 1000008
74: FR C = R11[0], R11 = R11 >> 1
if (FR C)
    69: R8 = R8 + R12, assign NZVC
    6B: if (V) R8 = ROR(R8) else R8 = ASR(R8), assign NZVC
else
    69: R8 = ASR(R8), assign NZVC
77: R9 = ROR(R9)
if (FR C == 0)
    75: FR C = R11[0], R11 = R11 >> 1
    69: next cycle
else
    75: R10 = R10, set FR N
if (FR N)
    76: R8 = R8 - R12
else MCIR = 1102
07: Rr = R8
06: R[r | 1] = R9
if (FR N==0 && FR Z==0)
    7C: PSW |= 0000018
    11: next instruction
else
    if (FR N==1)
        7C: R12 = R8 - 1777778
        if (FR Z==0)
            7E: PSW |= 0000018
            11: next instruction
        else
            7E: R9 = R9
            7D: PSW |= N ? 0000008 : 0000018
            11: next instruction
    else
        7C: R9 = R9
        if (FR N==0 && FR Z==0)
            7F: R12 = 0000018, set NZVC
            11: next instruction
        else
            if (FR N==0)
                7F: PSW |= 0000008
                11: next instruction
            else
                7F: PSW ^= 0000018, set NZVC
                11: next instruction

```



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 9. Набор процессорных инструкций

В данном разделе описаны процессорные инструкции, исполнение которых обеспечивает микропроцессор 1801BM1A, а также последовательности микроадресов и выполняемые внутренние действия микропрограммного автомата. При описании кодировки инструкций использованы следующие обозначения:

r	номер регистра общего назначения 0-7
dd	6-битовое поле адресации приемника, старшие 3 бита метод, младшие 3 бита номер регистра
ss	6-битовое поле адресации приемника, старшие 3 бита метод, младшие 3 бита номер регистра
x	Произвольная восьмеричная цифра
nzvc	4-битовое поле флагов в инструкции манипуляции арифметическими флагами
uu	6-битовое беззнаковое целое
bb	8-битовое знаковое поле, задает смещение условного перехода
nn	8-битовое беззнаковое поле

Кодирование методов адресации в полях источника и приемника

0	Rn		Прямая регистровая, операнд непосредственно в регистре Rn
1	@Rn		Косвенная, в регистре Rn находится адрес операнда
2	(Rn)+	#E	Автоинкрементная, в регистре Rn находится адрес операнда, после извлечения данных из памяти регистр Rn будет увеличен на 1 для байтовых операций и на 2 для словных операций
3	@(Rn)+	@#E	Автоинкрементная косвенная, в регистре Rn находится адрес указателя на операнд, после извлечения указателя из памяти регистр Rn будет увеличен на 2
4	-(Rn)		Автодекрементная, в регистре Rn находится адрес операнда, до извлечения данных из памяти регистр Rn уменьшается на 1 для байтовых операций и на 2 для словных операций
5	@-(Rn)		Автодекрементная косвенная, в регистре Rn находится адрес указателя на операнд, до извлечения указателя из памяти регистр Rn уменьшается на 2
6	E(Rn)	E	Индексная, адрес операнда вычисляется как сумма 16-битного смещения E и регистра Rn
7	@E(Rn)	@E	Индексная косвенная, адрес указателя на операнд вычисляется как сумма 16-битного смещения E и регистра Rn

Код инструкции	Мнемоника	NZVC	Описание
000000 <sub>8</sub>	HALT	*	Переход в пультовой режим. Код исключения записывается в MCIR и далее используется цепочка микрокода для обработки исключений пультового режима 01: MCIR = 011 <sub>2</sub> , VSEL = 1011 <sub>2</sub>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			01: core exception (160002 <sub>8</sub> )
000001 <sub>8</sub>	WAIT	-	<p>Ожидание прерывания, микропрограммный автомат циклически опрашивает блок прерываний, до возникновения прерывания или сброса бита <b>WCPU</b> записью в регистр управления внешним агентом.</p> <p><b>PC</b> указывает на следующую за WAIT инструкцию, если обработчик прерывания не изменит адрес возврата то исполнение продолжится со следующей за WAIT инструкции.</p> <p>Установленный бит <b>T</b> не приводит к немедленному выходу из режима ожидания, если нет активных незамаскированных аппаратных прерываний, позволяя выполняться инструкции WAIT предусмотренным способом</p> <p>01: PC = PC + 2  37: MCIR = 000<sub>2</sub>, set WCPU  01: MCIR poll  01: if (MCIR != 000<sub>2</sub>) exception/interrupt</p>
000002 <sub>8</sub>	RTI	*	<p>Возврат из прерывания. PC = (SP)+, PSW = (SP)+, старший байт <b>PSW</b> обнуляется, значение загруженное из стека игнорируется</p> <p>01: A = SP, word read start, SP = SP + 2  38: wait read, PC = Q  39: A = SP, read start, SP = SP + 2  6B: PSW = Q  2F: PSW = PSW &amp; ~177400<sub>8</sub>  if (T)  36: MCIR = 101<sub>2</sub>, VSEL = 0011<sub>2</sub>  01: user exception (000014<sub>8</sub>)  else  37: MCIR poll, A = PC  IR cmd load start, PC = PC + 2  01: wait read, next instruction</p>
000003 <sub>8</sub>	BPT	*	<p>Инструкция BPT, записывает код исключения непосредственно в <b>MCIR</b> и перестартует исполнение с адреса 01, опрос блока прерываний не запускается, начнется обработка исключения по вектору 000014<sub>8</sub> в пользовательском режиме</p> <p>01: MCIR = 101<sub>2</sub>, VSEL = 0011<sub>2</sub>  01: user exception (000014<sub>8</sub>)</p>
000004 <sub>8</sub>	IOT	*	<p>Инструкция IOT, записывает код исключения непосредственно в <b>MCIR</b> и перестартует исполнение с адреса 01, опрос блока прерываний не запускается, начнется обработка исключения по вектору 000020<sub>8</sub> в пользовательском режиме</p> <p>01: MCIR = 101<sub>2</sub>, VSEL = 0001<sub>2</sub>  01: user exception (000020<sub>8</sub>)</p>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
000005 <sub>8</sub>	RESET	-	<p>Сброс внешних устройств, на выходе <b>nINIT</b> формируется активный сигнал низкого уровня длительностью 224 такта <b>CLC</b>, затем сигнал деактивируется и снова выдерживается пауза 224 такта <b>CLC</b>. Данный сигнал также инициализирует ВЕ-таймер и детекторы фронтов внешних прерываний.</p> <p>Микропрограмма активирует сигнал <b>nINIT</b>, затем в <b>R12</b> записывается значение 340<sub>8</sub>, потом в цикле вычитается по 2, при достижении нулевого значения <b>nINIT</b> деактивируется, в <b>R12</b> снова заносится значение 340<sub>8</sub> и цикл вычитания повторяется, по достижении нулевого значения происходит переход к следующей инструкции</p> <pre> 01: MCIR = 010<sub>2</sub>, INIT set 36: R12 = 340<sub>8</sub> 37: while !(FR Z) R12 = R12 - 2 37: INIT reset, MCIR = 110<sub>2</sub> 36: 36: R12 = 340<sub>8</sub> 37: while !(FR Z) R12 = R12 - 2 11: next instruction </pre>
000006 <sub>8</sub>	RTT	*	<p>Возврат из прерывания. PC = (SP)+, PSW = (SP)+, старший байт <b>PSW</b> обнуляется, значение загруженное из стека игнорируется. После исполнения задерживает возникновение исключения по биту <b>T</b> (если он установлен в слове <b>PSW</b> извлекаемом из стека) на одну инструкцию</p> <pre> 01: A = SP, word read start, SP = SP + 2 38: wait read, PC = Q 39: A = SP, read start, SP = SP + 2 6B: PSW = Q 2F: PSW = PSW &amp; ~177400<sub>8</sub> if (T)     36: A = PC, IR cmd preload start,         PC = PC + 2     6A: wait read     01: next instruction else     37: A = PC, IR cmd load start,         PC = PC + 2     01: wait read, next instruction </pre>
000007 <sub>8</sub>	MFPT	-	<p>Недопустимая команда</p> <pre> 01: user exception (000010<sub>8</sub>) </pre>
000010 <sub>8</sub> 000011 <sub>8</sub> 000012 <sub>8</sub> 000013 <sub>8</sub>	START	*	<p>Возобновление исполнения пользовательской программы из пультового режима, <b>PC</b> и <b>PSW</b> берутся из вектора 177674<sub>8</sub>, бит 3 в регистре 177716<sub>8</sub> сбрасывается после чтения данных из вектора, так происходит переключение памяти в “пользовательский режим”</p>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			01: R12 = 177676 <sub>8</sub> , A = R12 2F: R12 = R12 - 2, A = R12, word read 2E: wait read, PC = Q 2D: R12 = R12 + 2, A = R12, word read 2C: wait read, PSW = Q 2B: R12 = 177716 <sub>8</sub> , A = R12, start rmw 2A: wait read, Q &= ~10 <sub>8</sub> , modified write 11: next instruction
000014 <sub>8</sub> 000015 <sub>8</sub> 000016 <sub>8</sub> 000017 <sub>8</sub>	STEP	*	Возобновление исполнения пользовательской программы из пультового режима, <b>PC</b> и <b>PSW</b> берутся из вектора 177674 <sub>8</sub> , бит 3 в регистре 177716 <sub>8</sub> сбрасывается после чтения данных из вектора, так происходит переключение памяти в “пользовательский режим”  01: R12 = 177676 <sub>8</sub> , A = R12 2F: R12 = R12 - 2, A = R12, word read 2E: wait read, PC = Q 2D: R12 = R12 + 2, A = R12, word read 2C: wait read, PSW = Q 2B: R12 = 177716 <sub>8</sub> , A = R12, start rmw 2A: wait read, Q &= ~10 <sub>8</sub> , modified write 7A: A = PC, word read, PC = PC + 2 6A: wait read 01: next instruction
00002x <sub>8</sub> 00003x <sub>8</sub> 00004x <sub>8</sub> 00005x <sub>8</sub> 00006x <sub>8</sub> 00007x <sub>8</sub>		-	Недопустимая команда  01: user exception (000010 <sub>8</sub> )
0001dd <sub>8</sub>	JMP dd	-	Безусловный переход, <b>PC</b> присваивается значение источника. Попытка исполнения инструкции JMP с кодом адресации приемника 0 записывает код исключения непосредственно в <b>MCIR</b> и перестартует исполнение с адреса 01, опрос блока прерываний не запускается, начнется обработка исключения по вектору 000004 <sub>8</sub> в пользовательском режиме  if (dd address mode == 0) 01: MCIR = 101 <sub>2</sub> , VSEL = 0100 <sub>2</sub> 01: user exception (000004 <sub>8</sub> ) else 01: MCIR = 100 <sub>2</sub> switch (dd address mode) case 0: Rd 45: user exception (000010 <sub>8</sub> ) case 1: @Rd 45: R12 = Rd case 2: (Rd)+

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			45: $Rd = Rd + 2$ 55: $R12 = Rd - 2$ case 3: $@(Rd) +$ 45: $A = Rd$ , word read start, $Rd = Rd + 2$ 4D: wait read, $R12 = Q$ case 4: $-(Rd)$ 45: $Rd = Rd - 2$ 55: $R12 = Rd$ case 5: $@-(Rd)$ 45: $Rd = Rd - 2$ , $A = Rd$ , word read start 4D: wait read, $R12 = Q$ case 6: $E(Rd)$ 45: $A = PC$ , word read start, $PC = PC + 2$ 4C: wait read, $R12 = Rd + Q$ case 7: $@E(Rd)$ 45: $A = PC$ , word read start, $PC = PC + 2$ 4C: wait read, $R12 = Rd + Q$ , $A = R12$ word read start 4D: wait read, $R12 = Q$ 48: $Q = Q$ 32: $PC = R12$ 11: next instruction
00020 <sub>8</sub>	RTS r	-	Возврат из подпрограммы. $PC = Rr$ , $Rr = *SP$ , $SP = SP + 2$ 01: $MCIR = 011_2$ 50: $R12 = Rr \& 17777_8$ 33: $PC = R12$ 15: $A = SP$ , word read start, $SP = SP + 2$ 40: wait read, $Rr = Q$ 11: next instruction
00021 <sub>8</sub> 00022 <sub>8</sub> 00023 <sub>8</sub>		-	Недопустимая команда 01: user exception (000010 <sub>8</sub> )
000240 <sub>8</sub>   nzvc	CL nzvc	*	Сбрасывает арифметические флаги в слове состояния <b>PSW</b> согласно установленным битам в поле <b>nzvc</b> кода операции
000240 <sub>8</sub> 000241 <sub>8</sub> 000242 <sub>8</sub> 000244 <sub>8</sub> 000250 <sub>8</sub> 000257 <sub>8</sub>	NOP CLN CLZ CLV CLC CCC	---- 0--- -0-- --0- ---0 0000	01: $A = PC$ , IR preload start 01: $PC = PC + 2$ , word read start 37: $R12 = IR[3:0]$ 36: $PSW = PSW \& \sim R12$ , MCIR poll 01: wait read, next instruction
000260 <sub>8</sub>   nzvc	SE nzvc	*	Устанавливает арифметические флаги в слове состояния <b>PSW</b> согласно установленным битам в поле <b>nzvc</b> кода операции
000260 <sub>8</sub> 000261 <sub>8</sub> 000262 <sub>8</sub>	NOP SEN SEZ	---- 1--- -1--	01: $A = PC$ , IR preload start 01: $PC = PC + 2$ , word read start 37: $R12 = IR[3:0]$

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
000264 <sub>8</sub> 000270 <sub>8</sub> 000277 <sub>8</sub>	SEV SEC SCC	--1- ---1 1111	36: PSW = PSW   R12, MCIR poll 01: wait read, next instruction
0003dd <sub>8</sub>	SWAB dd	**00	<p>Обмен байтов в слове приемника  tmp = dd, tmp = (tmp &lt;&lt; 8)   (tmp &gt;&gt; 8), dd = tmp,  N = dd[7], Z = (dd[7:0] == 0), C = 0, V = 0</p> <p>if (dd address mode == 0)  01: A = PC  IR preload start, word read start  if (dd register == PC)  01: PC = PC + 2  50: Rd = (Rd &gt;&gt; 8)   (Rd &lt;&lt; 8)  51: R12 = Rd &amp; ~177400<sub>8</sub>, set NZVC  11: next instruction  else  01: R12 = PC + 2  50: Rd = (Rd &gt;&gt; 8)   (Rd &lt;&lt; 8)  51: R12 = Rd &amp; ~177400<sub>8</sub>, assign NZVC,  MCIR poll, wait read  01: next instruction  01: MCIR = 111<sub>2</sub>  switch (dd address mode)  case 0: Rd  45: user exception (000010<sub>8</sub>)  case 1: @Rd  45: A = Rd, R12 = Rd, word rmw start  case 2: (Rd)+  45: A = Rd, Rd = Rd + 2, word rmw start  55: R12 = Rd - 2  case 3: @(Rd)+  45: A = Rd, word read start, Rd = Rd + 2  4D: wait read, R12 = Q,  A = Q, word rmw start  case 4: -(Rd)  45: Rd = Rd - 2, A = Rd, word rmw start  55: R12 = Rd  case 5: @-(Rd)  45: Rd = Rd - 2, A = Rd, word read start  4D: wait read, R12 = Q  A = Q, word rmw start  case 6: E(Rd)  45: A = PC, word read start, PC = PC + 2  4C: wait read, R12 = Rd + Q,  A = R12, word rmw start  case 7: @E(Rd)  45: A = PC, word read start, PC = PC + 2  4C: wait read, R12 = Rd + Q, A = R12  word read start</p>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			4D: wait read, R12 = Q, A = Rd, word rmw start 48: wait read, Q = (Q >> 8)   (Q << 8), modified data write 37: R12 = byte Q, assign NZVC 11: next instruction
0004bb <sub>8</sub>	BR bb	-	Безусловный переход в пределах -128..+128 слов от базового адреса инструкции ("."). 8-битовое поле смещения в коде инструкции умножается на 2 и знаково расширяется до 16 бит по старшему биту 01: MCIR = 010 <sub>2</sub> 7B: PC = PC + sign extended(IR[7:0]*2) 11: next instruction
0010bb <sub>8</sub>	BNE bb	-	Переход если Z==0 01: MCIR = 010 <sub>2</sub> 7B: if (Z==0) PC = PC + sign extended(IR[7:0]*2) else R12 = PC + sign extended(IR[7:0]*2) 11: next instruction
0014bb <sub>8</sub>	BEQ bb	-	Переход если Z==1 01: MCIR = 010 <sub>2</sub> 7B: if (Z) PC = PC + sign extended(IR[7:0]*2) else R12 = PC + sign extended(IR[7:0]*2) 11: next instruction
0020bb <sub>8</sub>	BGE bb	-	Переход если (N ^ V) == 0 01: MCIR = 010 <sub>2</sub> 7B: if !(N ^ V) PC = PC + sign extended(IR[7:0]*2) else R12 = PC + sign extended(IR[7:0]*2) 11: next instruction
0024bb <sub>8</sub>	BLT bb	-	Переход если (N ^ V) == 1 01: MCIR = 010 <sub>2</sub> 7B: if (N ^ V) PC = PC + sign extended(IR[7:0]*2) else R12 = PC + sign extended(IR[7:0]*2) 11: next instruction
0030bb <sub>8</sub>	BGT bb	-	Переход если (Z   (N ^ V)) == 0

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			01: MCIR = 010 <sub>2</sub> 7B: if !(Z   (N ^ V)) PC = PC + sign extended(IR[7:0]*2) else R12 = PC + sign extended(IR[7:0]*2) 11: next instruction
0034bb <sub>8</sub>	BLE bb	-	Переход если (Z   (N ^ V)) == 1 01: MCIR = 010 <sub>2</sub> 7B: if (Z   (N ^ V)) PC = PC + sign extended(IR[7:0]*2) else R12 = PC + sign extended(IR[7:0]*2) 11: next instruction
004rdd <sub>8</sub>	JSR r, dd	-	Вызов подпрограммы, указанный регистр сохраняется в стеке, <b>РС</b> записывается в регистр и затем получает значение приемника. Попытка исполнения инструкции JSR с кодом адресации приемника 0 записывает код исключения непосредственно в <b>MCIR</b> и перестартует исполнение с адреса 01, опрос блока прерываний не запускается, начнется обработка исключения по вектору 000004 <sub>8</sub> в пользовательском режиме If (dd address mode == 0) 01: MCIR = 101 <sub>2</sub> , VSEL = 0100 <sub>2</sub> ; 01: user exception (000004 <sub>8</sub> ) else 01: MCIR = 100 <sub>2</sub> switch (dd address mode) case 0: Rd 45: user exception (000010 <sub>8</sub> ) case 1: @Rd 45: R12 = Rd case 2: (Rd)+ 45: Rd = Rd + 2 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q case 4: -(Rd) 45: Rd = Rd - 2 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q case 7: @E(Rd)



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q 48: Q = Q 37: SP = SP - 2, A = SP, word write start 27: Rr = Rr & 177777 <sub>8</sub> , Q = Rr, write data 26: Rr = PC 32: PC = R12 11: next instruction
0050dd <sub>8</sub>	CLR dd	0100	<b>Присвоение приемнику нулевого значения</b> dd = 0, V = 0, C = 0, N = 0, Z = 1 if (dd address mode == 0) 01: A = PC IR preload start, word read start if (dd register == PC) 01: R12 = PC + 2 31: MCIR = 101 <sub>2</sub> 49: PC = 000000 <sub>8</sub> , assign NZVC 11: next instruction else 01: PC = PC + 2 31: MCIR = 101 <sub>2</sub> 49: Rd = 000000 <sub>8</sub> , assign NZVC, MCIR poll, wait read 01: next instruction else 01: MCIR = 111 <sub>2</sub> switch (dd address mode) case 0: Rd 45: user exception (000010 <sub>8</sub> ) case 1: @Rd 45: A = Rd, R12 = Rd, word rmw start case 2: (Rd)+ 45: A = Rd, Rd = Rd + 2, word rmw start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, word rmw start case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, word rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, word rmw start case 6: E(Rd)

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, word rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, word rmw start 48: wait read, Q = 000000 <sub>8</sub> , assign NZVC modified data write 11: next instruction
0051dd <sub>8</sub>	COM dd	**01	Побитовое инвертирование операнда dd = ~dd, V = 0, C = 0, N = dd[15], Z = (dd==0) if (dd address mode == 0) 01: A = PC IR preload start, word read start if (dd register == PC) 01: R12 = PC + 2 31: MCIR = 101 <sub>2</sub> 49: PC = PC ^ 177777 <sub>8</sub> , assign NZVC 11: next instruction else 01: PC = PC + 2 31: MCIR = 101 <sub>2</sub> 49: Rd = Rd ^ 177777 <sub>8</sub> , assign NZVC, MCIR poll, wait read 01: next instruction else 01: MCIR = 111 <sub>2</sub> switch (dd address mode) case 0: Rd 45: user exception (000010 <sub>8</sub> ) case 1: @Rd 45: A = Rd, R12 = Rd, word rmw start case 2: (Rd)+ 45: A = Rd, Rd = Rd + 2, word rmw start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, word rmw start case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, word rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<p>A = Q, word rmw start</p> <p>case 6: E(Rd)</p> <p>45: A = PC, word read start, PC = PC + 2</p> <p>4C: wait read, R12 = Rd + Q,</p> <p>A = R12, word rmw start</p> <p>case 7: @E(Rd)</p> <p>45: A = PC, word read start, PC = PC + 2</p> <p>4C: wait read, R12 = Rd + Q, A = R12</p> <p>word read start</p> <p>4D: wait read, R12 = Q,</p> <p>A = Rd, word rmw start</p> <p>48: wait read, Q = Q ^ 177777<sub>8</sub>, assign NZVC</p> <p>modified data write</p> <p>11: next instruction</p>
0052dd <sub>8</sub>	INC dd	***_	<p>Прибавление 1 к операнду</p> <p>dd = dd + 1, assign V, N = dd[15], Z = (dd==0)</p> <p>фактически V установится, если dd был равен 077777<sub>8</sub></p> <p>if (dd address mode == 0)</p> <p>01: A = PC</p> <p>IR preload start, word read start</p> <p>if (dd register == PC)</p> <p>01: R12 = PC + 2</p> <p>31: MCIR = 101<sub>2</sub></p> <p>49: PC = PC + 1, assign NZV</p> <p>11: next instruction</p> <p>else</p> <p>01: PC = PC + 2</p> <p>31: MCIR = 101<sub>2</sub></p> <p>49: Rd = Rd + 1, assign NZV,</p> <p>MCIR poll, wait read</p> <p>01: next instruction</p> <p>else</p> <p>01: MCIR = 111<sub>2</sub></p> <p>switch (dd address mode)</p> <p>case 0: Rd</p> <p>45: user exception (000010<sub>8</sub>)</p> <p>case 1: @Rd</p> <p>45: A = Rd, R12 = Rd, word rmw start</p> <p>case 2: (Rd)+</p> <p>45: A = Rd, Rd = Rd + 2, word rmw start</p> <p>55: R12 = Rd - 2</p> <p>case 3: @(Rd)+</p> <p>45: A = Rd, word read start, Rd = Rd + 2</p> <p>4D: wait read, R12 = Q,</p> <p>A = Q, word rmw start</p> <p>case 4: -(Rd)</p> <p>45: Rd = Rd - 2, A = Rd, word rmw start</p> <p>55: R12 = Rd</p>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, word rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, word rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, word rmw start 48: wait read, Q = Q + 000001 <sub>8</sub> , assign NZV modified data write 11: next instruction
0053dd <sub>8</sub>	DEC dd	***_	<b>Вычитание 1 из операнда</b> <b>dd = dd + 177777<sub>8</sub>, assign V, N = dd[15], Z = (dd==0)</b> <b>фактически V установится, если dd был равен 100000<sub>8</sub></b> if (dd address mode == 0) 01: A = PC IR preload start, word read start if (dd register == PC) 01: R12 = PC + 2 31: MCIR = 101 <sub>2</sub> 49: PC = PC - 1, assign NZV 11: next instruction else 01: PC = PC + 2 31: MCIR = 101 <sub>2</sub> 49: Rd = Rd - 1, assign NZV, MCIR poll, wait read 01: next instruction else 01: MCIR = 111 <sub>2</sub> switch (dd address mode) case 0: Rd 45: user exception (000010 <sub>8</sub> ) case 1: @Rd 45: A = Rd, R12 = Rd, word rmw start case 2: (Rd)+ 45: A = Rd, Rd = Rd + 2, word rmw start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, word rmw start

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, word rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, word rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, word rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, word rmw start 48: wait read, Q = Q - 000001 <sub>8</sub> , assign NZV modified data write 11: next instruction
0054dd <sub>8</sub>	NEG dd	****	Меняет знак операнда dd = ~dd + 1, assign CV, N = dd[15], Z = (dd==0) фактически V установится, если dd был равен 100000 <sub>8</sub> if (dd address mode == 0) 01: A = PC IR preload start, word read start if (dd register == PC) 01: R12 = PC + 2 31: MCIR = 101 <sub>2</sub> 49: PC = 0 - PC, assign NZVC 11: next instruction else 01: PC = PC + 2 31: MCIR = 101 <sub>2</sub> 49: Rd = 0 - Rd, assign NZVC, MCIR poll, wait read 01: next instruction else 01: MCIR = 111 <sub>2</sub> switch (dd address mode) case 0: Rd 45: user exception (000010 <sub>8</sub> ) case 1: @Rd 45: A = Rd, R12 = Rd, word rmw start case 2: (Rd)+ 45: A = Rd, Rd = Rd + 2, word rmw start 55: R12 = Rd - 2 case 3: @(Rd)+

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, word rmw start case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, word rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, word rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, word rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, word rmw start 48: wait read, Q = 000000 <sub>8</sub> - Q, assign NZVC modified data write 11: next instruction
0055dd <sub>8</sub>	ADC dd	****	<b>Прибавление к операнду флага переноса C</b> <b>dd = dd + C, assign CV, N = dd[15], Z = (dd==0)</b> <b>фактически V установится, если dd был равен 077777<sub>8</sub></b> if (dd address mode == 0) 01: A = PC IR preload start, word read start if (dd register == PC) 01: R12 = PC + 2 31: MCIR = 101 <sub>2</sub> 49: PC = PC + C, assign NZVC 11: next instruction else 01: PC = PC + 2 31: MCIR = 101 <sub>2</sub> 49: Rd = Rd + C, assign NZVC, MCIR poll, wait read 01: next instruction else 01: MCIR = 111 <sub>2</sub> switch (dd address mode) case 0: Rd 45: user exception (000010 <sub>8</sub> ) case 1: @Rd 45: A = Rd, R12 = Rd, word rmw start case 2: (Rd)+

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			45: A = Rd, Rd = Rd + 2, word rmw start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, word rmw start case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, word rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, word rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, word rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, word rmw start 48: wait read, Q = Q + C, assign NZVC modified data write 11: next instruction
0056dd <sub>8</sub>	SBC dd	****	<b>Вычитание из операнда флага переноса C</b> dd = dd - C, assign CV, N = dd[15], Z = (dd==0) if (dd address mode == 0) 01: A = PC IR preload start, word read start if (dd register == PC) 01: R12 = PC + 2 31: MCIR = 101 <sub>2</sub> 49: PC = PC - C, assign NZVC 11: next instruction else 01: PC = PC + 2 31: MCIR = 101 <sub>2</sub> 49: Rd = Rd - C, assign NZVC, MCIR poll, wait read 01: next instruction else 01: MCIR = 111 <sub>2</sub> switch (dd address mode) case 0: Rd 45: user exception (000010 <sub>8</sub> ) case 1: @Rd

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			45: A = Rd, R12 = Rd, word rmw start case 2: (Rd)+ 45: A = Rd, Rd = Rd + 2, word rmw start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, word rmw start case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, word rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, word rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, word rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, word rmw start 48: wait read, Q = Q - C, assign NZVC modified data write 11: next instruction
0057dd <sub>8</sub>	TST dd	**00	Установка флагов условий <b>PSW</b> соответственно операнду C = 0, V = 0, N = dd[15], Z = (dd==0) if (dd address mode == 0) 01: A = PC IR preload start, word read start if (dd register == PC) 01: R12 = PC + 2 31: MCIR = 101 <sub>2</sub> 49: PC = PC & 177777 <sub>8</sub> , assign NZVC 11: next instruction else 01: PC = PC + 2 31: MCIR = 101 <sub>2</sub> 49: Rd = Rd & 177777 <sub>8</sub> , assign NZVC, MCIR poll, wait read 01: next instruction else 01: MCIR = 101 <sub>2</sub> switch (dd address mode) case 0: Rd



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			45: user exception (000010 <sub>8</sub> ) case 1: @Rd 45: A = Rd, R12 = Rd, word read start case 2: (Rd)+ 45: A = Rd, Rd = Rd + 2, word read start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, word read start case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, word read start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, word read start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, word read start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, word read start 48: wait read, Q = Q & 17777 <sub>8</sub> , assign NZVC 11: next instruction
0060dd <sub>8</sub>	ROR dd	****	Кольцевой сдвиг вправо, вращает все биты операнда на одну позицию вправо, старший бит загружается из флага переноса, младший бит, выдвинутый из операнда, загружается во флаг переноса tmp = C, C = dd[0], dd = dd >> 1, dd[15] = tmp, N = dd[15], Z = (dd==0), V = C ^ N if (dd address mode == 0) 01: A = PC IR preload start, word read start if (dd register == PC) 01: R12 = PC + 2 31: MCIR = 101 <sub>2</sub> 49: PC = ROR(PC), assign NZVC 11: next instruction else 01: PC = PC + 2 31: MCIR = 101 <sub>2</sub> 49: Rd = ROR(Rd), assign NZVC, MCIR poll, wait read

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> 01: next instruction else 01: MCIR = 111<sub>2</sub> switch (dd address mode) case 0: Rd 45: user exception (000010<sub>8</sub>) case 1: @Rd 45: A = Rd, R12 = Rd, word rmw start case 2: (Rd)+ 45: A = Rd, Rd = Rd + 2, word rmw start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, word rmw start case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, word rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, word rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, word rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, word rmw start 48: wait read, Q = ROR(Q), assign NZVC modified data write 11: next instruction </pre>
0061dd <sub>8</sub>	ROL dd	****	<p>Кольцевой сдвиг влево, вращает все биты операнда на одну позицию влево, младший бит грузится из флага переноса, выдвинутый старший бит загружается во флаг переноса  tmp = C, C = dd[15], dd = dd &lt;&lt; 1, dd[0] = tmp,  N = dd[15], Z = (dd==0), V = C ^ N</p> <pre> if (dd address mode == 0) 01: A = PC IR preload start, word read start if (dd register == PC) 01: R12 = PC + 2 31: MCIR = 101<sub>2</sub> 49: PC = ROL(PC), assign NZVC 11: next instruction </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> else     01: PC = PC + 2     31: MCIR = 101<sub>2</sub>     49: Rd = ROL(Rd), assign NZVC,         MCIR poll, wait read     01: next instruction else     01: MCIR = 111<sub>2</sub>     switch (dd address mode)     case 0: Rd         45: user exception (000010<sub>8</sub>)     case 1: @Rd         45: A = Rd, R12 = Rd, word rmw start     case 2: (Rd)+         45: A = Rd, Rd = Rd + 2, word rmw start         55: R12 = Rd - 2     case 3: @(Rd)+         45: A = Rd, word read start, Rd = Rd + 2         4D: wait read, R12 = Q,             A = Q, word rmw start     case 4: -(Rd)         45: Rd = Rd - 2, A = Rd, word rmw start         55: R12 = Rd     case 5: @-(Rd)         45: Rd = Rd - 2, A = Rd, word read start         4D: wait read, R12 = Q             A = Q, word rmw start     case 6: E(Rd)         45: A = PC, word read start, PC = PC + 2         4C: wait read, R12 = Rd + Q,             A = R12, word rmw start     case 7: @E(Rd)         45: A = PC, word read start, PC = PC + 2         4C: wait read, R12 = Rd + Q, A = R12             word read start         4D: wait read, R12 = Q,             A = Rd, word rmw start     48: wait read, Q = ROL(Q), assign NZVC         modified data write     11: next instruction </pre>
0062dd <sub>8</sub>	ASR dd	****	<p>Арифметический сдвиг вправо, содержимое операнда сдвигается на одну позицию вправо, старший (знаковый) бит остается неизменным, флаг переноса грузится содержимым выдвинутого бита. Операцию можно трактовать как целочисленное деление знакового операнда на два, с остатком, остающимся во флаге переноса</p> <p><math>C = dd[0]</math>, <math>dd = dd \gg 1</math>, <math>dd[15] = dd[14]</math>,  <math>N = dd[15]</math>, <math>Z = (dd == 0)</math>, <math>V = C \wedge N</math></p>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> if (dd address mode == 0)   01: A = PC   IR preload start, word read start   if (dd register == PC)     01: R12 = PC + 2     31: MCIR = 101<sub>2</sub>     49: PC = ASR(PC), assign NZVC     11: next instruction   else     01: PC = PC + 2     31: MCIR = 101<sub>2</sub>     49: Rd = ASR(Rd), assign NZVC,       MCIR poll, wait read     01: next instruction else   01: MCIR = 111<sub>2</sub>   switch (dd address mode)     case 0: Rd       45: user exception (000010<sub>8</sub>)     case 1: @Rd       45: A = Rd, R12 = Rd, word rmw start     case 2: (Rd)+       45: A = Rd, Rd = Rd + 2, word rmw start       55: R12 = Rd - 2     case 3: @(Rd)+       45: A = Rd, word read start, Rd = Rd + 2       4D: wait read, R12 = Q,         A = Q, word rmw start     case 4: -(Rd)       45: Rd = Rd - 2, A = Rd, word rmw start       55: R12 = Rd     case 5: @-(Rd)       45: Rd = Rd - 2, A = Rd, word read start       4D: wait read, R12 = Q         A = Q, word rmw start     case 6: E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q,         A = R12, word rmw start     case 7: @E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q, A = R12         word read start       4D: wait read, R12 = Q,         A = Rd, word rmw start   48: wait read, Q = ASR(Q), assign NZVC       modified data write   11: next instruction </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
0063dd <sub>8</sub>	ASL dd	****	<p>Арифметический сдвиг влево, содержимое операнда сдвигается на одну позицию влево, младший бит обнуляется, выдвинутый знаковый бит грузится во флаг переноса. Операцию можно рассматривать как целочисленное умножение операнда на два</p> $C = dd[15], dd = dd \ll 1, N = dd[15], Z = (dd == 0), V = C \wedge N$ <pre> if (dd address mode == 0)   01: A = PC   IR preload start, word read start   if (dd register == PC)     01: R12 = PC + 2     31: MCIR = 101<sub>2</sub>     49: PC = ASL(PC), assign NZVC     11: next instruction   else     01: PC = PC + 2     31: MCIR = 101<sub>2</sub>     49: Rd = ASL(Rd), assign NZVC,       MCIR poll, wait read     01: next instruction else   01: MCIR = 111<sub>2</sub>   switch (dd address mode)     case 0: Rd       45: user exception (000010<sub>8</sub>)     case 1: @Rd       45: A = Rd, R12 = Rd, word rmw start     case 2: (Rd)+       45: A = Rd, Rd = Rd + 2, word rmw start       55: R12 = Rd - 2     case 3: @(Rd)+       45: A = Rd, word read start, Rd = Rd + 2       4D: wait read, R12 = Q,         A = Q, word rmw start     case 4: -(Rd)       45: Rd = Rd - 2, A = Rd, word rmw start       55: R12 = Rd     case 5: @-(Rd)       45: Rd = Rd - 2, A = Rd, word read start       4D: wait read, R12 = Q         A = Q, word rmw start     case 6: E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q,         A = R12, word rmw start     case 7: @E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q, A = R12 </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<p>word read start</p> <p>4D: wait read, R12 = Q,</p> <p>A = Rd, word rmw start</p> <p>48: wait read, Q = ASL(Q), assign NZVC</p> <p>modified data write</p> <p>11: next instruction</p>
0064nn <sub>8</sub>	MARK nn	-	<p>Удаление параметров из стека, SP = PC+2*nn, PC = R5, R5 = (SP)+; nn число параметров (6-битное беззнаковое поле)</p> <p>01: R12 = PC + IR[5:0]*2</p> <p>37: SP = R12</p> <p>36: PC = R5</p> <p>39: A = SP, SP = SP + 2, word read start</p> <p>6B: wait read, R5 = Q</p> <p>11: next instruction</p>
0065xx <sub>8</sub> 0066xx <sub>8</sub>	MFPI MTPI	-	<p>Недопустимая команда</p> <p>01: user exception (000010<sub>8</sub>)</p>
0067dd <sub>8</sub>	SXT dd	-*0-	<p>Распространение знака, устанавливает приемник в 0 или 177777<sub>8</sub> в зависимости от флага N</p> <p>dd = N ? 177777<sub>8</sub> : 0<sub>8</sub>, Z = N, V = 0</p> <p>if (dd address mode == 0)</p> <p>01: A = PC</p> <p>IR preload start, word read start</p> <p>if (dd register == PC)</p> <p>01: R12 = PC + 2</p> <p>31: MCIR = 101<sub>2</sub></p> <p>49: PC = N ? 177777<sub>8</sub>:0, assign NZV</p> <p>11: next instruction</p> <p>else</p> <p>01: PC = PC + 2</p> <p>31: MCIR = 101<sub>2</sub></p> <p>49: Rd = N ? 177777<sub>8</sub>:0, assign NZV,</p> <p>MCIR poll, wait read</p> <p>01: next instruction</p> <p>else</p> <p>01: MCIR = 111<sub>2</sub></p> <p>switch (dd address mode)</p> <p>case 0: Rd</p> <p>45: user exception (000010<sub>8</sub>)</p> <p>case 1: @Rd</p> <p>45: A = Rd, R12 = Rd, word rmw start</p> <p>case 2: (Rd)+</p> <p>45: A = Rd, Rd = Rd + 2, word rmw start</p> <p>55: R12 = Rd - 2</p> <p>case 3: @(Rd)+</p> <p>45: A = Rd, word read start, Rd = Rd + 2</p> <p>4D: wait read, R12 = Q,</p>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			A = Q, word rmw start case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, word rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, word rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, word rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, word rmw start 48: wait read, Q = N?177777 <sub>8</sub> :0, assign NZV modified data write 11: next instruction
007xxx <sub>8</sub>		-	Недопустимая команда 01: user exception (000010 <sub>8</sub> )
01ssdd <sub>8</sub>	MOV ss, dd	**0-	Присвоение приемнику значения источника dd = ss, V = 0, N = dd[15], Z = (dd==0) if (dd address mode == 0) && (ss address mode == 0) 01: A = PC IR preload start, word read start if (ss register == PC)    (dd register == PC) 01: R12 = PC + 2 58: Rd = Rs, assign NZV 11: next instruction else 01: PC = PC + 2 58: Rd = Rs, assign NZV MCIR poll, wait read else 01: MCIR = 011 <sub>2</sub> switch (ss address mode) case 0: Rs 47: R12 = Rs 5A: R11 = Rs case 1: @Rs 47: A = Rs, R12 = Rs, word read start 04: wait read, R11 = Q

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> case 2: (Rs)+   47: A = Rs, Rs = Rs + 2, word read start   04: wait read, R11 = Q case 3: @(Rs)+   47: A = Rs, word read start, Rs = Rs + 2   4E: wait read, R12 = Q,       A = Q, word read start   04: wait read, R11 = Q case 4: -(Rs)   47: Rs = Rs - 2, A = Rs, word read start   04: wait read, R11 = Q case 5: @-(Rs)   47: Rs = Rs - 2, A = Rs, word read start   4E: wait read, R12 = Q       A = Q, word read start   04: wait read, R11 = Q case 6: E(Rs)   47: A = PC, word read start, PC = PC + 2   46: wait read, R12 = Rs + Q,       A = R12, word read start   04: wait read, R11 = Q case 7: @E(Rs)   47: A = PC, word read start, PC = PC + 2   46: wait read, R12 = Rs + Q, A = R12       word read start   4E: wait read, R12 = Q,       A = Q, word read start   04: wait read, R11 = Q  if (dd address mode == 0)   if (dd register == 7)     30: A = PC, R12 = PC+2, IR preload start         word read start     59: Rd = R11, assign NVZ     11: next instruction   else     30: A = PC, PC = PC + 2, IR preload start         word read start     59: Rd = R11, assign NVZ, MCIR poll     01: next instruction   else     switch(dd address mode)     case 0: Rd       45: user exception (000010<sub>8</sub>)     case 1: @Rd       45: A = Rd, R12 = Rd, word write start     case 2: (Rd)+       45: A = Rd, Rd = Rd + 2, word write       55: R12 = Rd - 2 </pre>



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, word write start case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, word write 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, word write start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, word write start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, word write start 48: R12 = R11, Q = R11, data write 19: Q = Q, assign NZV 11: next instruction </pre>
02ssddg	CMP ss, dd	****	<p>Вычисление разности (ss – dd) без сохранения результата в приемнике, установка флагов <b>PSW</b> по результатам вычислений. Операция не изменяет исходных операндов. Порядок вычитания отличается от команды SUB.</p> <p>tmp = ss + (~dd + 1), assign CV, N = tmp[15], Z = (tmp==0)</p> <pre> if (dd address mode == 0) &amp;&amp; (ss address mode == 0) 01: A = PC IR preload start, word read start if (ss register == PC)    (dd register == PC) 01: R12 = PC + 2 58: R12 = Rs - Rd, assign NZVC 11: next instruction else 01: PC = PC + 2 58: R12 = Rs - Rd, assign NZVC MCIR poll, wait read else 01: MCIR = 001<sub>2</sub> switch (ss address mode) case 0: Rs 47: R12 = Rs 5A: R11 = Rs </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> case 1: @Rs   47: A = Rs, R12 = Rs, word read start   04: wait read, R11 = Q case 2: (Rs)+   47: A = Rs, Rs = Rs + 2, word read start   04: wait read, R11 = Q case 3: @(Rs)+   47: A = Rs, word read start, Rs = Rs + 2   4E: wait read, R12 = Q,       A = Q, word read start   04: wait read, R11 = Q case 4: -(Rs)   47: Rs = Rs - 2, A = Rs, word read start   04: wait read, R11 = Q case 5: @-(Rs)   47: Rs = Rs - 2, A = Rs, word read start   4E: wait read, R12 = Q       A = Q, word read start   04: wait read, R11 = Q case 6: E(Rs)   47: A = PC, word read start, PC = PC + 2   46: wait read, R12 = Rs + Q,       A = R12, word read start   04: wait read, R11 = Q case 7: @E(Rs)   47: A = PC, word read start, PC = PC + 2   46: wait read, R12 = Rs + Q, A = R12       word read start   4E: wait read, R12 = Q,       A = Q, word read start   04: wait read, R11 = Q  if (dd address mode == 0)   if (dd register == 7)     30: A = PC, R12 = PC+2, IR preload start         word read start     59: Rd = R11, assign NVZ     11: next instruction   else     30: A = PC, PC = PC + 2, IR preload start         word read start     59: Rd = R11, assign NVZ, MCIR poll     01: next instruction   else     switch(dd address mode)     case 0: Rd       45: user exception (000010<sub>8</sub>)     case 1: @Rd       45: A = Rd, R12 = Rd, word read start </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			case 2: (Rd)+ 45: A = Rd, Rd = Rd + 2, word read 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, word read start case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, word read 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, word read start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, word read start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, word read start 48: wait read, R12 = R11 - Q, assign NZVC 19: Q = Q, assign NZV 11: next instruction
03ssddg	BIT ss, dd	**0-	<b>Проверка битов приемника по маске источника</b> <b>tmp = ss &amp; dd, V = 0, N = tmp[15], Z = (tmp==0)</b>  if (dd address mode == 0) && (ss address mode == 0) 01: A = PC IR preload start, word read start if (ss register == PC)    (dd register == PC) 01: R12 = PC + 2 58: R12 = Rs & Rd, assign NZV 11: next instruction else 01: PC = PC + 2 58: R12 = Rs & Rd, assign NZV MCIR poll, wait read else 01: MCIR = 001 <sub>2</sub> switch (ss address mode) case 0: Rs 47: R12 = Rs 5A: R11 = Rs

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> case 1: @Rs   47: A = Rs, R12 = Rs, word read start   04: wait read, R11 = Q case 2: (Rs)+   47: A = Rs, Rs = Rs + 2, word read start   04: wait read, R11 = Q case 3: @(Rs)+   47: A = Rs, word read start, Rs = Rs + 2   4E: wait read, R12 = Q,       A = Q, word read start   04: wait read, R11 = Q case 4: -(Rs)   47: Rs = Rs - 2, A = Rs, word read start   04: wait read, R11 = Q case 5: @-(Rs)   47: Rs = Rs - 2, A = Rs, word read start   4E: wait read, R12 = Q       A = Q, word read start   04: wait read, R11 = Q case 6: E(Rs)   47: A = PC, word read start, PC = PC + 2   46: wait read, R12 = Rs + Q,       A = R12, word read start   04: wait read, R11 = Q case 7: @E(Rs)   47: A = PC, word read start, PC = PC + 2   46: wait read, R12 = Rs + Q, A = R12       word read start   4E: wait read, R12 = Q,       A = Q, word read start   04: wait read, R11 = Q  if (dd address mode == 0)   if (dd register == 7)     30: A = PC, R12 = PC+2, IR preload start         word read start     59: Rd = R11, assign NVZ     11: next instruction   else     30: A = PC, PC = PC + 2, IR preload start         word read start     59: Rd = R11, assign NVZ, MCIR poll     01: next instruction else   switch(dd address mode)   case 0: Rd     45: user exception (000010<sub>8</sub>)   case 1: @Rd     45: A = Rd, R12 = Rd, word read start </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			case 2: (Rd)+ 45: A = Rd, Rd = Rd + 2, word read 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, word read start case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, word read 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, word read start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, word read start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, word read start 48: wait read, R12 = R11 & Q, assign NZV 19: Q = Q, assign NZV 11: next instruction
04ssddg	BIC ss, dd	**0-	<b>Сброс битов приемника по маске источника</b> <b>dd = dd &amp; ~ss, V = 0, N = dd[15], Z = (dd==0)</b>  if (dd address mode == 0) && (ss address mode == 0) 01: A = PC IR preload start, word read start if (ss register == PC)    (dd register == PC) 01: R12 = PC + 2 58: Rd = Rd & ~Rs, assign NZV 11: next instruction else 01: PC = PC + 2 58: Rd = Rd & ~Rs, assign NZV MCIR poll, wait read else 01: MCIR = 011 <sub>2</sub> switch (ss address mode) case 0: Rs 47: R12 = Rs 5A: R11 = Rs

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> case 1: @Rs   47: A = Rs, R12 = Rs, word read start   04: wait read, R11 = Q case 2: (Rs)+   47: A = Rs, Rs = Rs + 2, word read start   04: wait read, R11 = Q case 3: @(Rs)+   47: A = Rs, word read start, Rs = Rs + 2   4E: wait read, R12 = Q,       A = Q, word read start   04: wait read, R11 = Q case 4: -(Rs)   47: Rs = Rs - 2, A = Rs, word read start   04: wait read, R11 = Q case 5: @-(Rs)   47: Rs = Rs - 2, A = Rs, word read start   4E: wait read, R12 = Q       A = Q, word read start   04: wait read, R11 = Q case 6: E(Rs)   47: A = PC, word read start, PC = PC + 2   46: wait read, R12 = Rs + Q,       A = R12, word read start   04: wait read, R11 = Q case 7: @E(Rs)   47: A = PC, word read start, PC = PC + 2   46: wait read, R12 = Rs + Q, A = R12       word read start   4E: wait read, R12 = Q,       A = Q, word read start   04: wait read, R11 = Q  if (dd address mode == 0)   if (dd register == 7)     30: A = PC, R12 = PC+2, IR preload start         word read start     59: Rd = R11, assign NVZ     11: next instruction   else     30: A = PC, PC = PC + 2, IR preload start         word read start     59: Rd = R11, assign NVZ, MCIR poll     01: next instruction   else     switch(dd address mode)     case 0: Rd       45: user exception (000010<sub>8</sub>)     case 1: @Rd       45: A = Rd, R12 = Rd, word rmw start </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> case 2: (Rd)+   45: A = Rd, Rd = Rd + 2, word rmw start   55: R12 = Rd - 2 case 3: @(Rd)+   45: A = Rd, word read start, Rd = Rd + 2   4D: wait read, R12 = Q,       A = Q, word rmw start case 4: -(Rd)   45: Rd = Rd - 2, A = Rd, word rmw start   55: R12 = Rd case 5: @-(Rd)   45: Rd = Rd - 2, A = Rd, word read start   4D: wait read, R12 = Q       A = Q, word rmw start case 6: E(Rd)   45: A = PC, word read start, PC = PC + 2   4C: wait read, R12 = Rd + Q,       A = R12, word rmw start case 7: @E(Rd)   45: A = PC, word read start, PC = PC + 2   4C: wait read, R12 = Rd + Q, A = R12       word rmw start   4D: wait read, R12 = Q,       A = Rd, word rmw start 48: wait read, R12 = Q &amp; ~R11, assign NZV 19: Q = Q, write modified, data, assign NZV 11: next instruction </pre>
05ssddg	BIS ss, dd	**0-	<p>Установка битов приемника по маске источника  <math>dd = dd   ss, V = 0, N = dd[15], Z = (dd == 0)</math></p> <pre> if (dd address mode == 0) &amp;&amp;   (ss address mode == 0) 01: A = PC   IR preload start, word read start   if (ss register == PC)        (dd register == PC)     01: R12 = PC + 2     58: Rd = Rd   Rs, assign NZV     11: next instruction   else     01: PC = PC + 2     58: Rd = Rd   Rs, assign NZV     MCIR poll, wait read else 01: MCIR = 011<sub>2</sub> switch (ss address mode) case 0: Rs   47: R12 = Rs   5A: R11 = Rs </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> case 1: @Rs   47: A = Rs, R12 = Rs, word read start   04: wait read, R11 = Q case 2: (Rs)+   47: A = Rs, Rs = Rs + 2, word read start   04: wait read, R11 = Q case 3: @(Rs)+   47: A = Rs, word read start, Rs = Rs + 2   4E: wait read, R12 = Q,       A = Q, word read start   04: wait read, R11 = Q case 4: -(Rs)   47: Rs = Rs - 2, A = Rs, word read start   04: wait read, R11 = Q case 5: @-(Rs)   47: Rs = Rs - 2, A = Rs, word read start   4E: wait read, R12 = Q       A = Q, word read start   04: wait read, R11 = Q case 6: E(Rs)   47: A = PC, word read start, PC = PC + 2   46: wait read, R12 = Rs + Q,       A = R12, word read start   04: wait read, R11 = Q case 7: @E(Rs)   47: A = PC, word read start, PC = PC + 2   46: wait read, R12 = Rs + Q, A = R12       word read start   4E: wait read, R12 = Q,       A = Q, word read start   04: wait read, R11 = Q  if (dd address mode == 0)   if (dd register == 7)     30: A = PC, R12 = PC+2, IR preload start         word read start     59: Rd = R11, assign NVZ     11: next instruction   else     30: A = PC, PC = PC + 2, IR preload start         word read start     59: Rd = R11, assign NVZ, MCIR poll     01: next instruction   else     switch(dd address mode)     case 0: Rd       45: user exception (000010<sub>8</sub>)     case 1: @Rd       45: A = Rd, R12 = Rd, word rmw start </pre>



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			case 2: (Rd)+ 45: A = Rd, Rd = Rd + 2, word rmw start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, word rmw start case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, word rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, word rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, word rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word rmw start 4D: wait read, R12 = Q, A = Rd, word rmw start 48: wait read, R12 = Q   R11, assign NZV 19: Q = Q, write modified, data, assign NZV 11: next instruction
06ssddg	ADD ss, dd	****	Добавление источника к приемнику dd = ss + dd, assign CV, N = dd[15], Z = (dd==0) if (dd address mode == 0) && (ss address mode == 0) 01: A = PC IR preload start, word read start if (ss register == PC)    (dd register == PC) 01: R12 = PC + 2 58: Rd = Rd + Rs, assign NZVC 11: next instruction else 01: PC = PC + 2 58: Rd = Rd + Rs, assign NZVC MCIR poll, wait read else 01: MCIR = 010 <sub>2</sub> switch (ss address mode) case 0: Rs 47: R12 = Rs 5A: R11 = Rs

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> case 1: @Rs   47: A = Rs, R12 = Rs, word read start   04: wait read, R11 = Q case 2: (Rs)+   47: A = Rs, Rs = Rs + 2, word read start   04: wait read, R11 = Q case 3: @(Rs)+   47: A = Rs, word read start, Rs = Rs + 2   4E: wait read, R12 = Q,       A = Q, word read start   04: wait read, R11 = Q case 4: -(Rs)   47: Rs = Rs - 2, A = Rs, word read start   04: wait read, R11 = Q case 5: @-(Rs)   47: Rs = Rs - 2, A = Rs, word read start   4E: wait read, R12 = Q       A = Q, word read start   04: wait read, R11 = Q case 6: E(Rs)   47: A = PC, word read start, PC = PC + 2   46: wait read, R12 = Rs + Q,       A = R12, word read start   04: wait read, R11 = Q case 7: @E(Rs)   47: A = PC, word read start, PC = PC + 2   46: wait read, R12 = Rs + Q, A = R12       word read start   4E: wait read, R12 = Q,       A = Q, word read start   04: wait read, R11 = Q  if (dd address mode == 0)   if (dd register == 7)     30: A = PC, R12 = PC+2, IR preload start         word read start     59: Rd = R11, assign NVZ     11: next instruction   else     30: A = PC, PC = PC + 2, IR preload start         word read start     59: Rd = R11, assign NVZ, MCIR poll     01: next instruction   else     switch(dd address mode)     case 0: Rd       45: user exception (000010<sub>8</sub>)     case 1: @Rd       45: A = Rd, R12 = Rd, word rmw start </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			case 2: (Rd)+ 45: A = Rd, Rd = Rd + 2, word rmw start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, word rmw start case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, word rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, word rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, word rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word rmw start 4D: wait read, R12 = Q, A = Rd, word rmw start 48: wait read, R12 = Q + R11, assign NZVC 19: Q = Q, write modified, data, assign NZVC 11: next instruction
070xxx <sub>8</sub> 071xxx <sub>8</sub> 072xxx <sub>8</sub> 073xxx <sub>8</sub>	MUL DIV ASH ASHC	-	Недопустимая команда 01: user exception (000010 <sub>8</sub> )
074rdd <sub>8</sub>	XOR r, dd	**0-	Исключающее ИЛИ, в качестве источника допускается только регистр, нет байтовой версии инструкции dd = dd ^ Rr V = 0, N = dd[15], Z = (dd==0) if (dd address mode == 0) 01: A = PC IR preload start, word read start if (dd register == PC)    (r register == PC) 01: R12 = PC + 2 58: Rd = Rr ^ Rd, assign NZVC 11: next instruction else 01: PC = PC + 2 58: Rd = Rr ^ Rd, assign NZVC, MCIR poll, wait read 01: next instruction

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> else   01: MCIR = 011<sub>2</sub>   5A: R11 = Rr   switch (dd address mode)   case 0: Rd     45: user exception (000010<sub>8</sub>)   case 1: @Rd     45: A = Rd, R12 = Rd, word rmw start   case 2: (Rd)+     45: A = Rd, Rd = Rd + 2, word rmw start     55: R12 = Rd - 2   case 3: @(Rd)+     45: A = Rd, word read start, Rd = Rd + 2     4D: wait read, R12 = Q,         A = Q, word rmw start   case 4: -(Rd)     45: Rd = Rd - 2, A = Rd, word rmw start     55: R12 = Rd   case 5: @-(Rd)     45: Rd = Rd - 2, A = Rd, word read start     4D: wait read, R12 = Q         A = Q, word rmw start   case 6: E(Rd)     45: A = PC, word read start, PC = PC + 2     4C: wait read, R12 = Rd + Q,         A = R12, word rmw start   case 7: @E(Rd)     45: A = PC, word read start, PC = PC + 2     4C: wait read, R12 = Rd + Q, A = R12         word read start     4D: wait read, R12 = Q,         A = Rd, word rmw start   48: wait read, Q = Q ^ R11, assign NZVC         modified data write   11: next instruction </pre>
075xxx <sub>8</sub> 076xxx <sub>8</sub>		-	<p>Недопустимая команда</p> <pre> 01: user exception (000010<sub>8</sub>) </pre>
077ruu <sub>8</sub>	SOB r, bb	-	<p>Вычитать 1 из регистра единицу и выполнить переход, если результат вычитания ненулевой. Переход возможен только назад в пределах 63 слов от адреса “.+2” (на который указывает <b>РС</b> в момент исполнения команды), никакие флаги <b>PSW</b> не изменяются</p> <pre> 01: GOTO 27 27: Rr = Rr - 1, set FR register 37: if (FR Z) R12 = PC - 2*IR[5:0];     else PC = PC - 2*IR[5:0]; 11: next instruction </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
1000bb <sub>8</sub>	BPL bb	-	<p>Переход если N==0</p> <pre> 01: MCIR = 010<sub>2</sub> 7B: if (!N)     PC = PC + sign extended(IR[7:0]*2)     else     R12 = PC + sign extended(IR[7:0]*2) 11: next instruction </pre>
1004bb <sub>8</sub>	BMI bb	-	<p>Переход если N==1</p> <pre> 01: MCIR = 010<sub>2</sub> 7B: if (N)     PC = PC + sign extended(IR[7:0]*2)     else     R12 = PC + sign extended(IR[7:0]*2) 11: next instruction </pre>
1010bb <sub>8</sub>	BHI bb	-	<p>Переход если (Z   C)==0</p> <pre> 01: MCIR = 010<sub>2</sub> 7B: if !(Z   C)     PC = PC + sign extended(IR[7:0]*2)     else     R12 = PC + sign extended(IR[7:0]*2) 11: next instruction </pre>
1014bb <sub>8</sub>	BLOS bb	-	<p>Переход если (Z   C)==1</p> <pre> 01: MCIR = 010<sub>2</sub> 7B: if (Z   C)     PC = PC + sign extended(IR[7:0]*2)     else     R12 = PC + sign extended(IR[7:0]*2) 11: next instruction </pre>
1020bb <sub>8</sub>	BVC bb	-	<p>Переход если V==0</p> <pre> 01: MCIR = 010<sub>2</sub> 7B: if !(V)     PC = PC + sign extended(IR[7:0]*2)     else     R12 = PC + sign extended(IR[7:0]*2) 11: next instruction </pre>
1024bb <sub>8</sub>	BVS bb	-	<p>Переход если V==1</p> <pre> 01: MCIR = 010<sub>2</sub> 7B: if (V)     PC = PC + sign extended(IR[7:0]*2)     else     R12 = PC + sign extended(IR[7:0]*2) 11: next instruction </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
1030bb <sub>8</sub>	BCC bb BHIS bb	-	<p>Переход если C==0</p> <pre> 01: MCIR = 010<sub>2</sub> 7B: if !(C)     PC = PC + sign extended(IR[7:0]*2)     else     R12 = PC + sign extended(IR[7:0]*2) 11: next instruction </pre>
1034bb <sub>8</sub>	BCS bb BLO bb	-	<p>Переход если C==1</p> <pre> 01: MCIR = 010<sub>2</sub> 7B: if (C)     PC = PC + sign extended(IR[7:0]*2)     else     R12 = PC + sign extended(IR[7:0]*2) 11: next instruction </pre>
1040nn <sub>8</sub>	EMT nn	*	<p>Инструкция EMT, записывает код исключения непосредственно в <b>MCIR</b> и перестартует исполнение с адреса 01, опрос блока прерываний не запускается, начнется обработка исключения по вектору 000030<sub>8</sub> в пользовательском режиме</p> <pre> 01: MCIR = 101<sub>2</sub>, VSEL = 0110<sub>2</sub>; 01: user exception (000030<sub>8</sub>) </pre>
1044nn <sub>8</sub>	TRAP nn	*	<p>Инструкция TRAP, записывает код исключения непосредственно в <b>MCIR</b> и перестартует исполнение с адреса 01, опрос блока прерываний не запускается, начнется обработка исключения по вектору 000034<sub>8</sub> в пользовательском режиме</p> <pre> 01: MCIR = 101<sub>2</sub>, VSEL = 1100<sub>2</sub>; 01: user exception (000034<sub>8</sub>) </pre>
1050dd <sub>8</sub>	CLRB dd	0100	<p>Присвоение приемнику нулевого значения dd = 0, V = 0, C = 0, N = 0, Z = 1</p> <pre> if (dd address mode == 0)     01: A = PC     IR preload start, word read start     if (dd register == PC)         01: R12 = PC + 2         31: MCIR = 101<sub>2</sub>         49: PC[7:0] = 000<sub>8</sub>, assign NZVC         11: next instruction     else         01: PC = PC + 2         31: MCIR = 101<sub>2</sub>         49: Rd[7:0] = 000<sub>8</sub>, assign NZVC,             MCIR poll, wait read         01: next instruction else     01: MCIR = 111<sub>2</sub> </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			switch (dd address mode) case 0: Rd 45: user exception (000010 <sub>8</sub> ) case 1: @Rd 45: A = Rd, R12 = Rd, byte rmw start case 2: (Rd)+ 45: A = Rd, Rd = Rd + 1, byte rmw start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, byte rmw start case 4: -(Rd) 45: Rd = Rd - 1, A = Rd, byte rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, byte rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, byte rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, byte rmw start 48: wait read, Q = 000000 <sub>8</sub> , assign NZVC modified byte data write 11: next instruction
1051dd <sub>8</sub>	COMB dd	**01	Побитовое инвертирование операнда dd = ~dd, V = 0, C = 0, N = dd[7], Z = (dd==0) if (dd address mode == 0) 01: A = PC IR preload start, word read start if (dd register == PC) 01: R12 = PC + 2 31: MCIR = 101 <sub>2</sub> 49: PC[7:0] ^= 377 <sub>8</sub> , assign NZVC 11: next instruction else 01: PC = PC + 2 31: MCIR = 101 <sub>2</sub> 49: Rd[7:0] ^= 377 <sub>8</sub> , assign NZVC, MCIR poll, wait read 01: next instruction

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> else   01: MCIR = 111<sub>2</sub>   switch (dd address mode)     case 0: Rd       45: user exception (000010<sub>8</sub>)     case 1: @Rd       45: A = Rd, R12 = Rd, byte rmw start     case 2: (Rd)+       45: A = Rd, Rd = Rd + 1, byte rmw start       55: R12 = Rd - 2     case 3: @(Rd)+       45: A = Rd, word read start, Rd = Rd + 2       4D: wait read, R12 = Q,         A = Q, byte rmw start     case 4: -(Rd)       45: Rd = Rd - 1, A = Rd, byte rmw start       55: R12 = Rd     case 5: @-(Rd)       45: Rd = Rd - 2, A = Rd, word read start       4D: wait read, R12 = Q         A = Q, byte rmw start     case 6: E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q,         A = R12, byte rmw start     case 7: @E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q, A = R12         word read start       4D: wait read, R12 = Q,         A = Rd, byte rmw start   48: wait read, Q ^= 177777<sub>8</sub>, assign NZVC       modified byte data write   11: next instruction </pre>
1052dd <sub>8</sub>	INCB dd	***_	<p>Прибавление 1 к операнду  dd = dd + 1, assign V, N = dd[7], Z = (dd==0)  фактически V установится если dd был равен 177<sub>8</sub></p> <pre> if (dd address mode == 0)   01: A = PC   IR preload start, word read start   if (dd register == PC)     01: R12 = PC + 2     31: MCIR = 101<sub>2</sub>     49: PC[7:0] += 1<sub>8</sub>, assign NZV     11: next instruction   else     01: PC = PC + 2     31: MCIR = 101<sub>2</sub> </pre>



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> 49: Rd[7:0] += 1, assign NZV,     MCIR poll, wait read 01: next instruction else 01: MCIR = 111<sub>2</sub> switch (dd address mode) case 0: Rd     45: user exception (000010<sub>8</sub>) case 1: @Rd     45: A = Rd, R12 = Rd, byte rmw start case 2: (Rd)+     45: A = Rd, Rd = Rd + 1, byte rmw start     55: R12 = Rd - 2 case 3: @(Rd)+     45: A = Rd, word read start, Rd = Rd + 2     4D: wait read, R12 = Q,         A = Q, byte rmw start case 4: -(Rd)     45: Rd = Rd - 1, A = Rd, byte rmw start     55: R12 = Rd case 5: @-(Rd)     45: Rd = Rd - 2, A = Rd, word read start     4D: wait read, R12 = Q         A = Q, byte rmw start case 6: E(Rd)     45: A = PC, word read start, PC = PC + 2     4C: wait read, R12 = Rd + Q,         A = R12, byte rmw start case 7: @E(Rd)     45: A = PC, word read start, PC = PC + 2     4C: wait read, R12 = Rd + Q, A = R12         word read start     4D: wait read, R12 = Q,         A = Rd, byte rmw start 48: wait read, Q = Q + 1, assign NZV     modified byte data write 11: next instruction </pre>
1053dd <sub>8</sub>	DECB dd	***-	<p>Вычитание 1 из операнда  dd = dd - 1, assign V, N = dd[7], Z = (dd==0)  фактически V установится если dd был равен 200<sub>8</sub></p> <pre> if (dd address mode == 0) 01: A = PC     IR preload start, word read start if (dd register == PC) 01: R12 = PC + 2 31: MCIR = 101<sub>2</sub> 49: PC[7:0] -= 1<sub>8</sub>, assign NZV 11: next instruction </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> else     01: PC = PC + 2     31: MCIR = 101<sub>2</sub>     49: Rd[7:0] -= 1, assign NZV,         MCIR poll, wait read     01: next instruction else     01: MCIR = 111<sub>2</sub>     switch (dd address mode)     case 0: Rd         45: user exception (000010<sub>8</sub>)     case 1: @Rd         45: A = Rd, R12 = Rd, byte rmw start     case 2: (Rd)+         45: A = Rd, Rd = Rd + 1, byte rmw start         55: R12 = Rd - 2     case 3: @(Rd)+         45: A = Rd, word read start, Rd = Rd + 2         4D: wait read, R12 = Q,             A = Q, byte rmw start     case 4: -(Rd)         45: Rd = Rd - 1, A = Rd, byte rmw start         55: R12 = Rd     case 5: @-(Rd)         45: Rd = Rd - 2, A = Rd, word read start         4D: wait read, R12 = Q             A = Q, byte rmw start     case 6: E(Rd)         45: A = PC, word read start, PC = PC + 2         4C: wait read, R12 = Rd + Q,             A = R12, byte rmw start     case 7: @E(Rd)         45: A = PC, word read start, PC = PC + 2         4C: wait read, R12 = Rd + Q, A = R12             word read start         4D: wait read, R12 = Q,             A = Rd, byte rmw start     48: wait read, Q = Q - 1, assign NZV         modified byte data write     11: next instruction </pre>
1054dd <sub>8</sub>	NEGB dd	****	<p>Меняет знак операнда  dd = ~dd + 1, assign CV, N = dd[7], Z = (dd==0)  фактически V установится если dd был равен 200<sub>8</sub></p> <pre> if (dd address mode == 0)     01: A = PC         IR preload start, word read start     if (dd register == PC)         01: R12 = PC + 2 </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> 31: MCIR = 101<sub>2</sub> 49: PC[7:0] = 0-PC[7:0], assign NZVC 11: next instruction else 01: PC = PC + 2 31: MCIR = 101<sub>2</sub> 49: Rd[7:0] = 0-Rd[7:0], assign NZVC,     MCIR poll, wait read 01: next instruction else 01: MCIR = 111<sub>2</sub> switch (dd address mode) case 0: Rd 45: user exception (000010<sub>8</sub>) case 1: @Rd 45: A = Rd, R12 = Rd, byte rmw start case 2: (Rd)+ 45: A = Rd, Rd = Rd + 1, byte rmw start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q,     A = Q, byte rmw start case 4: -(Rd) 45: Rd = Rd - 1, A = Rd, byte rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q     A = Q, byte rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q,     A = R12, byte rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12     word read start 4D: wait read, R12 = Q,     A = Rd, byte rmw start 48: wait read, Q = 0 - Q, assign NZVC     modified byte data write 11: next instruction </pre>
1055dd <sub>8</sub>	ADCB dd	****	<p>Прибавление к операнду флага переноса C  dd = dd + C, assign CV, N = dd[7], Z = (dd==0)  фактически V установится если dd был равен 0177<sub>8</sub></p> <pre> if (dd address mode == 0) 01: A = PC </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> IR preload start, word read start if (dd register == PC)     01: R12 = PC + 2     31: MCIR = 101<sub>2</sub>     49: PC[7:0] += C, assign NZVC     11: next instruction else     01: PC = PC + 2     31: MCIR = 101<sub>2</sub>     49: Rd[7:0] += C, assign NZVC,         MCIR poll, wait read     01: next instruction else     01: MCIR = 111<sub>2</sub>     switch (dd address mode)         case 0: Rd             45: user exception (000010<sub>8</sub>)         case 1: @Rd             45: A = Rd, R12 = Rd, byte rmw start         case 2: (Rd)+             45: A = Rd, Rd = Rd + 1, byte rmw start             55: R12 = Rd - 2         case 3: @(Rd)+             45: A = Rd, word read start, Rd = Rd + 2             4D: wait read, R12 = Q,                 A = Q, byte rmw start         case 4: -(Rd)             45: Rd = Rd - 1, A = Rd, byte rmw start             55: R12 = Rd         case 5: @-(Rd)             45: Rd = Rd - 2, A = Rd, word read start             4D: wait read, R12 = Q                 A = Q, byte rmw start         case 6: E(Rd)             45: A = PC, word read start, PC = PC + 2             4C: wait read, R12 = Rd + Q,                 A = R12, byte rmw start         case 7: @E(Rd)             45: A = PC, word read start, PC = PC + 2             4C: wait read, R12 = Rd + Q, A = R12                 word read start             4D: wait read, R12 = Q,                 A = Rd, byte rmw start     48: wait read, Q = Q + C, assign NZVC         modified byte data write     11: next instruction </pre>
1056dd <sub>8</sub>	SBCB dd	****	<p><b>Вычитание из операнда флага переноса C</b>  dd = dd - C, assign CV, N = dd[7], Z = (dd==0)</p>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> if (dd address mode == 0)   01: A = PC   IR preload start, word read start   if (dd register == PC)     01: R12 = PC + 2     31: MCIR = 101<sub>2</sub>     49: PC[7:0] -= C, assign NZVC     11: next instruction   else     01: PC = PC + 2     31: MCIR = 101<sub>2</sub>     49: Rd[7:0] -= C, assign NZVC,       MCIR poll, wait read     01: next instruction else   01: MCIR = 111<sub>2</sub>   switch (dd address mode)     case 0: Rd       45: user exception (000010<sub>8</sub>)     case 1: @Rd       45: A = Rd, R12 = Rd, byte rmw start     case 2: (Rd)+       45: A = Rd, Rd = Rd + 1, byte rmw start       55: R12 = Rd - 2     case 3: @(Rd)+       45: A = Rd, word read start, Rd = Rd + 2       4D: wait read, R12 = Q,         A = Q, byte rmw start     case 4: -(Rd)       45: Rd = Rd - 1, A = Rd, byte rmw start       55: R12 = Rd     case 5: @-(Rd)       45: Rd = Rd - 2, A = Rd, word read start       4D: wait read, R12 = Q         A = Q, byte rmw start     case 6: E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q,         A = R12, byte rmw start     case 7: @E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q, A = R12         word read start       4D: wait read, R12 = Q,         A = Rd, byte rmw start   48: wait read, Q = Q - C, assign NZVC       modified byte data write   11: next instruction </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
1057dd <sub>8</sub>	TSTB dd	**00	<p>Установка флагов условий <b>PSW</b> соответственно операнду C = 0, V = 0, N = dd[7], Z = (dd==0)</p> <pre> if (dd address mode == 0)   01: A = PC   IR preload start, word read start   if (dd register == PC)     01: R12 = PC + 2     31: MCIR = 101<sub>2</sub>     49: PC = PC &amp; 177777<sub>8</sub>, assign NZVC     11: next instruction   else     01: PC = PC + 2     31: MCIR = 101<sub>2</sub>     49: Rd = Rd &amp; 177777<sub>8</sub>, assign NZVC,       MCIR poll, wait read     01: next instruction else   01: MCIR = 101<sub>2</sub>   switch (dd address mode)     case 0: Rd       45: user exception (000010<sub>8</sub>)     case 1: @Rd       45: A = Rd, R12 = Rd, word read start     case 2: (Rd)+       45: A = Rd, Rd = Rd + 2, word read start       55: R12 = Rd - 2     case 3: @(Rd)+       45: A = Rd, word read start, Rd = Rd + 2       4D: wait read, R12 = Q,         A = Q, word read start     case 4: -(Rd)       45: Rd = Rd - 2, A = Rd, word read start       55: R12 = Rd     case 5: @-(Rd)       45: Rd = Rd - 2, A = Rd, word read start       4D: wait read, R12 = Q         A = Q, word read start     case 6: E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q,         A = R12, word read start     case 7: @E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q, A = R12         word read start       4D: wait read, R12 = Q,         A = Rd, word read start       48: wait read, Q = Q &amp; 177777<sub>8</sub>, assign NZVC </pre>

Однокристальный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			11: next instruction
1060dd <sub>8</sub>	RORB dd	****	<p>Кольцевой сдвиг вправо, вращает все биты операнда на одну позицию вправо, старший бит загружается из флага переноса, младший бит, выдвинутый из операнда, загружается во флаг переноса</p> <p>tmp = C, C = dd[0], dd = dd &gt;&gt; 1, dd[7] = tmp, N = dd[7], Z = (dd==0), V = C ^ N</p> <p>if (dd address mode == 0)  01: A = PC  IR preload start, word read start  if (dd register == PC)  01: R12 = PC + 2  31: MCIR = 101<sub>2</sub>  49: PC[7:0] = RORB(PC), assign NZVC  11: next instruction  else  01: PC = PC + 2  31: MCIR = 101<sub>2</sub>  49: Rd[7:0] = RORB(Rd), assign NZVC,  MCIR poll, wait read  01: next instruction  else  01: MCIR = 111<sub>2</sub>  switch (dd address mode)  case 0: Rd  45: user exception (000010<sub>8</sub>)  case 1: @Rd  45: A = Rd, R12 = Rd, byte rmw start  case 2: (Rd)+  45: A = Rd, Rd = Rd + 1, byte rmw start  55: R12 = Rd - 2  case 3: @(Rd)+  45: A = Rd, word read start, Rd = Rd + 2  4D: wait read, R12 = Q,  A = Q, byte rmw start  case 4: -(Rd)  45: Rd = Rd - 1, A = Rd, byte rmw start  55: R12 = Rd  case 5: @-(Rd)  45: Rd = Rd - 2, A = Rd, word read start  4D: wait read, R12 = Q  A = Q, byte rmw start  case 6: E(Rd)  45: A = PC, word read start, PC = PC + 2  4C: wait read, R12 = Rd + Q,  A = R12, byte rmw start  case 7: @E(Rd)  45: A = PC, word read start, PC = PC + 2</p>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, byte rmw start 48: wait read, Q = RORB(Q), assign NZVC modified byte data write 11: next instruction
1061dd <sub>8</sub>	ROLB dd	****	<p>Кольцевой сдвиг влево, вращает все биты операнда на одну позицию влево, младший бит грузится из флага переноса, выдвинутый старший бит загружается во флаг переноса  tmp = C, C = dd[7], dd = dd &lt;&lt; 1, dd[0] = tmp,  N = dd[7], Z = (dd==0), V = C ^ N</p> <pre> if (dd address mode == 0) 01: A = PC IR preload start, word read start if (dd register == PC) 01: R12 = PC + 2 31: MCIR = 101<sub>2</sub> 49: PC[7:0] = ROLB(PC), assign NZVC 11: next instruction else 01: PC = PC + 2 31: MCIR = 101<sub>2</sub> 49: Rd[7:0] = ROLB(Rd), assign NZVC, MCIR poll, wait read 01: next instruction else 01: MCIR = 111<sub>2</sub> switch (dd address mode) case 0: Rd 45: user exception (000010<sub>8</sub>) case 1: @Rd 45: A = Rd, R12 = Rd, byte rmw start case 2: (Rd)+ 45: A = Rd, Rd = Rd + 1, byte rmw start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, byte rmw start case 4: -(Rd) 45: Rd = Rd - 1, A = Rd, byte rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, byte rmw start case 6: E(Rd) </pre>



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, byte rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, byte rmw start 48: wait read, Q = ROLB(Q), assign NZVC modified byte data write 11: next instruction
1062dd <sub>8</sub>	ASRB dd	****	<p>Арифметический сдвиг вправо, содержимое операнда сдвигается на одну позицию вправо, старший (знаковый) бит остается неизменным, флаг переноса грузится содержимым выдвинутого бита. Операцию можно трактовать как целочисленное деление знакового операнда на два, с остатком, остающимся во флаге переноса</p> $C = dd[0], dd = dd \gg 1, dd[7] = dd[6],$ $N = dd[7], Z = (dd == 0), V = C \wedge N$ <pre> if (dd address mode == 0) 01: A = PC IR preload start, word read start if (dd register == PC) 01: R12 = PC + 2 31: MCIR = 101<sub>2</sub> 49: PC[7:0] = ASRB(PC), assign NZVC 11: next instruction else 01: PC = PC + 2 31: MCIR = 101<sub>2</sub> 49: Rd[7:0] = ASRB(Rd), assign NZVC, MCIR poll, wait read 01: next instruction else 01: MCIR = 111<sub>2</sub> switch (dd address mode) case 0: Rd 45: user exception (000010<sub>8</sub>) case 1: @Rd 45: A = Rd, R12 = Rd, byte rmw start case 2: (Rd)+ 45: A = Rd, Rd = Rd + 1, byte rmw start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, byte rmw start </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> case 4: -(Rd) 45: Rd = Rd - 1, A = Rd, byte rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, byte rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, byte rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, byte rmw start 48: wait read, Q = ASRB(Q), assign NZVC modified byte data write 11: next instruction </pre>
1063dd <sub>8</sub>	ASLB dd	****	<p>Арифметический сдвиг влево, содержимое операнда сдвигается на одну позицию влево, младший бит обнуляется, выдвинутый знаковый бит грузится во флаг переноса. Операцию можно рассматривать как целочисленное умножение операнда на два</p> <p><math>C = dd[7], dd = dd \ll 1, N = dd[7], Z = (dd == 0), V = C \wedge N</math></p> <pre> if (dd address mode == 0) 01: A = PC IR preload start, word read start if (dd register == PC) 01: R12 = PC + 2 31: MCIR = 101<sub>2</sub> 49: PC[7:0] = ASLB(PC), assign NZVC 11: next instruction else 01: PC = PC + 2 31: MCIR = 101<sub>2</sub> 49: Rd[7:0] = ASLB(Rd), assign NZVC, MCIR poll, wait read 01: next instruction else 01: MCIR = 111<sub>2</sub> switch (dd address mode) case 0: Rd 45: user exception (000010<sub>8</sub>) case 1: @Rd 45: A = Rd, R12 = Rd, byte rmw start case 2: (Rd)+ </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> 45: A = Rd, Rd = Rd + 1, byte rmw start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q,     A = Q, byte rmw start case 4: -(Rd) 45: Rd = Rd - 1, A = Rd, byte rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q     A = Q, byte rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q,     A = R12, byte rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12     word read start 4D: wait read, R12 = Q,     A = Rd, byte rmw start 48: wait read, Q = ASLB(Q), assign NZVC     modified byte data write 11: next instruction </pre>
1064ss <sub>8</sub>	MTPS ss	*	<p>Запись источника в слово состояния процессора <b>PSW</b>, позволяет задать только младшие 8 бит <b>PSW</b>, бит <b>T</b> не изменяется. Операндом является байт – автоинкремент и автодекремент адреса на единицу.</p> <pre> if (dd address mode == 0)     01: A = PC, IR preload start,         word read start, R12 = PC + 2     50: Rd = Rd &amp; ~000020<sub>8</sub>, Q = Rd     37: PSW[7:0] = PSW[7:0] &amp; 000020<sub>8</sub>     36: PSW[7:0] = PSW[7:0]   Q     11: next instruction else     01: MCIR = 101<sub>2</sub>     switch (dd address mode)         case 0: Rd             45: user exception (000010<sub>8</sub>)         case 1: @Rd             45: A = Rd, R12 = Rd, byte read start         case 2: (Rd)+             45: A = Rd, Rd = Rd + 1, byte read start             55: R12 = Rd - 1         case 3: @(Rd)+ </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, byte read start case 4: -(Rd) 45: Rd = Rd - 1, A = Rd, byte read start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, byte read start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, byte read start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, byte start 48: wait read, Q = Q & ~000020 <sub>8</sub> 37: PSW[7:0] = PSW[7:0] & 000020 <sub>8</sub> 36: PSW[7:0] = PSW[7:0]   Q 11: next instruction
1065xx <sub>8</sub> 1066xx <sub>8</sub>	MFPD MTPD	-	Недопустимая команда 01: user exception (000010 <sub>8</sub> )
1067dd <sub>8</sub>	MFPS dd	**0-	Запись в приемник слова состояния процессора <b>PSW</b> , трактуется как байтовая, если приемником является регистр то старший байт заполняется знаковым расширением (как MOVb) if (dd address mode == 0) 01: A = PC IR preload start, word read start if (dd register == PC) 01: R12 = PC + 2 50: PC = PSW 51: if (N) PC  = 177400 <sub>8</sub> else PC &= ~177400 <sub>8</sub> assign NZV 11: next instruction else 01: PC = PC + 2 50: Rd = PSW 51: if (N) Rd  = 177400 <sub>8</sub> else Rd &= ~177400 <sub>8</sub> assign NZV, MCIR poll, wait read 01: next instruction else

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			01: MCIR = 111 <sub>2</sub> switch (dd address mode) case 0: Rd 45: user exception (000010 <sub>8</sub> ) case 1: @Rd 45: A = Rd, R12 = Rd, byte rmw start case 2: (Rd)+ 45: A = Rd, Rd = Rd + 1, byte rmw start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, byte rmw start case 4: -(Rd) 45: Rd = Rd - 1, A = Rd, byte rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, byte rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, byte rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, byte rmw start 48: wait read, Q = PSW, assign NZV modified byte data write 11: next instruction
107xxx <sub>8</sub>		-	Недопустимая команда 01: user exception (000010 <sub>8</sub> )
11ssdd <sub>8</sub>	MOVB ss, dd	**0-	Присвоение приемнику значения источника, Если приемник является регистром, то старший байт принимает значение знакового расширения младшего байта dd = ss, V = 0, N = dd[7], Z = (dd[7:0]==0), if (dd addressing mode == 0) dd[15:8] = dd[7] ? 377 <sub>8</sub> : 0  if (dd address mode == 0) && (ss address mode == 0) 01: A = PC IR preload start, word read start if (ss register == PC)    (dd register == PC) 01: R12 = PC + 2

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> 58: Rd = Rs 51: if (N) Rd  = 177400<sub>8</sub>     else Rd &amp;= ~177400<sub>8</sub>     assign NZV 11: next instruction else 01: PC = PC + 2 58: Rd = Rs 51: if (N) Rd  = 177400<sub>8</sub>     else Rd &amp;= ~177400<sub>8</sub>     assign NZV, MCIR poll, wait read 01: next instruction else 01: MCIR = 011<sub>2</sub> switch (ss address mode) case 0: Rs 47: R12 = Rs 5A: R11 = Rs case 1: @Rs 47: A = Rs, R12 = Rs, byte read start 04: wait read, R11 = Q case 2: (Rs)+ 47: A = Rs, Rs = Rs + 2, byte read start 04: wait read, R11 = Q case 3: @(Rs)+ 47: A = Rs, word read start, Rs = Rs + 2 4E: wait read, R12 = Q,     A = Q, byte read start 04: wait read, R11 = Q case 4: -(Rs) 47: Rs = Rs - 2, A = Rs, byte read start 04: wait read, R11 = Q case 5: @-(Rs) 47: Rs = Rs - 2, A = Rs, word read start 4E: wait read, R12 = Q     A = Q, byte read start 04: wait read, R11 = Q case 6: E(Rs) 47: A = PC, word read start, PC = PC + 2 46: wait read, R12 = Rs + Q,     A = R12, byte read start 04: wait read, R11 = Q case 7: @E(Rs) 47: A = PC, word read start, PC = PC + 2 46: wait read, R12 = Rs + Q, A = R12     word read start 4E: wait read, R12 = Q,     A = Q, byte read start 04: wait read, R11 = Q </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> if (dd address mode == 0)   if (dd register == 7)     30: A = PC, R12 = PC + 2,       IR preload start, word read start     59: Rd = R11     51: if (N) Rd  = 177400<sub>8</sub>       else Rd &amp;= ~177400<sub>8</sub>       assign NZV, MCIR poll, wait read     11: next instruction   else     30: A = PC, PC = PC + 2, IR preload start       word read start     59: Rd = R11     51: if (N) Rd  = 177400<sub>8</sub>       else Rd &amp;= ~177400<sub>8</sub>       assign NZV, MCIR poll, wait read     01: next instruction   else     switch(dd address mode)     case 0: Rd       45: user exception (000010<sub>8</sub>)     case 1: @Rd       45: A = Rd, R12 = Rd, byte write start     case 2: (Rd)+       45: A = Rd, Rd = Rd + 2, byte write       55: R12 = Rd - 2     case 3: @(Rd)+       45: A = Rd, word read start, Rd = Rd + 2       4D: wait read, R12 = Q,       A = Q, byte write start     case 4: -(Rd)       45: Rd = Rd - 2, A = Rd, byte write       55: R12 = Rd     case 5: @-(Rd)       45: Rd = Rd - 2, A = Rd, word read start       4D: wait read, R12 = Q       A = Q, byte write start     case 6: E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q,       A = R12, byte write start     case 7: @E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q, A = R12       byte read start       4D: wait read, R12 = Q,       A = Rd, word write start     48: R12 = R11, Q = R11, data byte write </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			19: Q = Q, assign NZV 11: next instruction
12ssddg	CMPB ss, dd	****	<p>Вычисление разности (ss – dd) без сохранения результата в приемнике, установка флагов <b>PSW</b> по результатам вычислений. Операция не изменяет исходных операндов. Порядок вычитания отличается от команды SUB.</p> <p>tmp = ss[7:0] – dd[7:0], assign CV, N = tmp[7], Z = (tmp==0)</p> <pre> if (dd address mode == 0) &amp;&amp; (ss address mode == 0) 01: A = PC IR preload start, word read start if (ss register == PC)    (dd register == PC) 01: R12 = PC + 2 58: R12[7:0] = Rs[7:0] - Rd[7:0], assign NZVC 11: next instruction else 01: PC = PC + 2 58: R12[7:0] = Rs[7:0] - Rd[7:0], assign NZVC, MCIR poll, wait read else 01: MCIR = 001<sub>2</sub> switch (ss address mode) case 0: Rs 47: R12 = Rs 5A: R11 = Rs case 1: @Rs 47: A = Rs, R12 = Rs, byte read start 04: wait read, R11 = Q case 2: (Rs)+ 47: A = Rs, Rs = Rs + 2, byte read start 04: wait read, R11 = Q case 3: @(Rs)+ 47: A = Rs, word read start, Rs = Rs + 2 4E: wait read, R12 = Q, A = Q, byte read start 04: wait read, R11 = Q case 4: -(Rs) 47: Rs = Rs - 2, A = Rs, byte read start 04: wait read, R11 = Q case 5: @-(Rs) 47: Rs = Rs - 2, A = Rs, word read start 4E: wait read, R12 = Q A = Q, byte read start 04: wait read, R11 = Q case 6: E(Rs) 47: A = PC, word read start, PC = PC + 2 </pre>



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> 46: wait read, R12 = Rs + Q,     A = R12, byte read start 04: wait read, R11 = Q case 7: @E(Rs) 47: A = PC, word read start, PC = PC + 2 46: wait read, R12 = Rs + Q, A = R12     word read start 4E: wait read, R12 = Q,     A = Q, byte read start 04: wait read, R11 = Q  if (dd address mode == 0)   if (dd register == 7)     30: A = PC, R12 = PC+2, IR preload start         word read start     59: R12 = Rs[7:0] - Rd[7:0], assign NZVC     11: next instruction   else     30: A = PC, PC = PC + 2, IR preload start         word read start     59: R12 = Rs[7:0] - Rd[7:0],         assign NZVC, MCIR poll     01: next instruction   else     switch(dd address mode)     case 0: Rd       45: user exception (000010<sub>8</sub>)     case 1: @Rd       45: A = Rd, R12 = Rd, byte read start     case 2: (Rd)+       45: A = Rd, Rd = Rd + 2, byte read       55: R12 = Rd - 2     case 3: @(Rd)+       45: A = Rd, word read start, Rd = Rd + 2       4D: wait read, R12 = Q,           A = Q, byte read start     case 4: -(Rd)       45: Rd = Rd - 2, A = Rd, byte read       55: R12 = Rd     case 5: @-(Rd)       45: Rd = Rd - 2, A = Rd, word read start       4D: wait read, R12 = Q           A = Q, byte read start     case 6: E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q,           A = R12, byte read start     case 7: @E(Rd)       45: A = PC, word read start, PC = PC + 2 </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, byte read start 48: wait read, Q = R11[7:0] - Q[7:0], assign NZVC 19: Q = Q, assign NZV 11: next instruction
13ssddg	BITB ss, dd	**0-	<b>Проверка битов приемника по маске источника</b> <b>tmp = ss &amp; dd, V = 0, N = tmp[7], Z = (tmp==0)</b> if (dd address mode == 0) && (ss address mode == 0) 01: A = PC IR preload start, word read start if (ss register == PC)    (dd register == PC) 01: R12 = PC + 2 58: R12[7:0] = Rs[7:0] & Rd[7:0], assign NZV 11: next instruction else 01: PC = PC + 2 58: R12[7:0] = Rs[7:0] & Rd[7:0], assign NZV, MCIR poll, wait read else 01: MCIR = 001 <sub>2</sub> switch (ss address mode) case 0: Rs 47: R12 = Rs 5A: R11 = Rs case 1: @Rs 47: A = Rs, R12 = Rs, byte read start 04: wait read, R11 = Q case 2: (Rs)+ 47: A = Rs, Rs = Rs + 2, byte read start 04: wait read, R11 = Q case 3: @(Rs)+ 47: A = Rs, word read start, Rs = Rs + 2 4E: wait read, R12 = Q, A = Q, byte read start 04: wait read, R11 = Q case 4: -(Rs) 47: Rs = Rs - 2, A = Rs, byte read start 04: wait read, R11 = Q case 5: @-(Rs) 47: Rs = Rs - 2, A = Rs, word read start 4E: wait read, R12 = Q A = Q, byte read start

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> 04: wait read, R11 = Q case 6: E(Rs) 47: A = PC, word read start, PC = PC + 2 46: wait read, R12 = Rs + Q,     A = R12, byte read start 04: wait read, R11 = Q case 7: @E(Rs) 47: A = PC, word read start, PC = PC + 2 46: wait read, R12 = Rs + Q, A = R12     word read start 4E: wait read, R12 = Q,     A = Q, byte read start 04: wait read, R11 = Q  if (dd address mode == 0)   if (dd register == 7)     30: A = PC, R12 = PC+2, IR preload start         word read start     59: R12 = Rs[7:0] - Rd[7:0], assign NZVC     11: next instruction   else     30: A = PC, PC = PC + 2, IR preload start         word read start     59: R12 = Rs[7:0] &amp; Rd[7:0],         assign NZVC, MCIR poll     01: next instruction   else     switch(dd address mode)     case 0: Rd       45: user exception (000010<sub>8</sub>)     case 1: @Rd       45: A = Rd, R12 = Rd, byte read start     case 2: (Rd)+       45: A = Rd, Rd = Rd + 2, byte read       55: R12 = Rd - 2     case 3: @(Rd)+       45: A = Rd, word read start, Rd = Rd + 2       4D: wait read, R12 = Q,           A = Q, byte read start     case 4: -(Rd)       45: Rd = Rd - 2, A = Rd, byte read       55: R12 = Rd     case 5: @-(Rd)       45: Rd = Rd - 2, A = Rd, word read start       4D: wait read, R12 = Q           A = Q, byte read start     case 6: E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q, </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> A = R12, byte read start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word read start 4D: wait read, R12 = Q, A = Rd, byte read start 48: wait read, Q = R11[7:0] &amp; Q[7:0], assign NZVC 19: Q = Q, assign NZV 11: next instruction </pre>
14ssddg	BICB ss, dd	**0-	<p><b>Сброс битов приемника по маске источника</b>  <b>dd = dd &amp; ~ss, V = 0, N = dd[7], Z = (dd==0)</b></p> <pre> if (dd address mode == 0) &amp;&amp; (ss address mode == 0) 01: A = PC IR preload start, word read start if (ss register == PC)    (dd register == PC) 01: R12 = PC + 2 58: Rd[7:0] = Rs[7:0] &amp; ~Rd[7:0], assign NZV 11: next instruction else 01: PC = PC + 2 58: Rd[7:0] = Rs[7:0] &amp; ~Rd[7:0], assign NZV, MCIR poll, wait read else 01: MCIR = 011<sub>2</sub> switch (ss address mode) case 0: Rs 47: R12 = Rs 5A: R11 = Rs case 1: @Rs 47: A = Rs, R12 = Rs, byte read start 04: wait read, R11 = Q case 2: (Rs)+ 47: A = Rs, Rs = Rs + 2, byte read start 04: wait read, R11 = Q case 3: @(Rs)+ 47: A = Rs, word read start, Rs = Rs + 2 4E: wait read, R12 = Q, A = Q, byte read start 04: wait read, R11 = Q case 4: -(Rs) 47: Rs = Rs - 2, A = Rs, byte read start 04: wait read, R11 = Q case 5: @-(Rs) </pre>

Однокристальный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> 47: Rs = Rs - 2, A = Rs, word read start 4E: wait read, R12 = Q     A = Q, byte read start 04: wait read, R11 = Q case 6: E(Rs) 47: A = PC, word read start, PC = PC + 2 46: wait read, R12 = Rs + Q,     A = R12, byte read start 04: wait read, R11 = Q case 7: @E(Rs) 47: A = PC, word read start, PC = PC + 2 46: wait read, R12 = Rs + Q, A = R12     word read start 4E: wait read, R12 = Q,     A = Q, byte read start 04: wait read, R11 = Q  if (dd address mode == 0) switch(dd address mode) case 0: Rd 45: user exception (000010<sub>8</sub>) case 1: @Rd 45: A = Rd, R12 = Rd, byte rmw start case 2: (Rd)+ 45: A = Rd, Rd = Rd + 2, byte rmw 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q,     A = Q, byte rmw start case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, byte rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q     A = Q, byte rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q,     A = R12, byte rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12     word read start 4D: wait read, R12 = Q,     A = Rd, byte rmw start 48: wait read, Q = R11[7:0] &amp; ~Q[7:0],     assign NZV, write modified byte data </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			19: Q = Q, assign NZV 11: next instruction
15ssddg	BISB ss, dd	**0-	<p>Установка битов приемника по маске источника  dd = dd   ss, V = 0, N = dd[7], Z = (dd==0)</p> <pre> if (dd address mode == 0) &amp;&amp; (ss address mode == 0) 01: A = PC IR preload start, word read start if (ss register == PC)    (dd register == PC) 01: R12 = PC + 2 58: Rd[7:0] = Rs[7:0]   Rd[7:0], assign NZV 11: next instruction else 01: PC = PC + 2 58: Rd[7:0] = Rs[7:0]   Rd[7:0], assign NZV, MCIR poll, wait read else 01: MCIR = 011<sub>2</sub> switch (ss address mode) case 0: Rs 47: R12 = Rs 5A: R11 = Rs case 1: @Rs 47: A = Rs, R12 = Rs, byte read start 04: wait read, R11 = Q case 2: (Rs)+ 47: A = Rs, Rs = Rs + 2, byte read start 04: wait read, R11 = Q case 3: @(Rs)+ 47: A = Rs, word read start, Rs = Rs + 2 4E: wait read, R12 = Q, A = Q, byte read start 04: wait read, R11 = Q case 4: -(Rs) 47: Rs = Rs - 2, A = Rs, byte read start 04: wait read, R11 = Q case 5: @-(Rs) 47: Rs = Rs - 2, A = Rs, word read start 4E: wait read, R12 = Q A = Q, byte read start 04: wait read, R11 = Q case 6: E(Rs) 47: A = PC, word read start, PC = PC + 2 46: wait read, R12 = Rs + Q, A = R12, byte read start 04: wait read, R11 = Q </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> case 7: @E(Rs)   47: A = PC, word read start, PC = PC + 2   46: wait read, R12 = Rs + Q, A = R12       word read start   4E: wait read, R12 = Q,       A = Q, byte read start   04: wait read, R11 = Q  if (dd address mode == 0)   switch(dd address mode)     case 0: Rd       45: user exception (000010<sub>8</sub>)     case 1: @Rd       45: A = Rd, R12 = Rd, byte rmw start     case 2: (Rd)+       45: A = Rd, Rd = Rd + 2, byte rmw       55: R12 = Rd - 2     case 3: @(Rd)+       45: A = Rd, word read start, Rd = Rd + 2       4D: wait read, R12 = Q,           A = Q, byte rmw start     case 4: -(Rd)       45: Rd = Rd - 2, A = Rd, byte rmw start 55: R12 = Rd     case 5: @-(Rd)       45: Rd = Rd - 2, A = Rd, word read start       4D: wait read, R12 = Q           A = Q, byte rmw start     case 6: E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q,           A = R12, byte rmw start     case 7: @E(Rd)       45: A = PC, word read start, PC = PC + 2       4C: wait read, R12 = Rd + Q, A = R12           word read start       4D: wait read, R12 = Q,           A = Rd, byte rmw start   48: wait read, Q = R11[7:0]   Q[7:0],       assign NZV, write modified byte data   19: Q = Q, assign NZV   11: next instruction </pre>
16ssdd <sub>8</sub>	SUB ss, dd	****	<p>Вычитание источника из приемника  dd = dd - ss, assign CV, N = dd[15], Z = (dd==0)</p> <pre> if (dd address mode == 0) &amp;&amp;   (ss address mode == 0)   01: A = PC       IR preload start, word read start </pre>

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> if (ss register == PC)    (dd register == PC) 01: R12 = PC + 2 58: Rd = Rd - Rs, assign NZVC 11: next instruction else 01: PC = PC + 2 58: Rd = Rd - Rs, assign NZVC MCIR poll, wait read else 01: MCIR = 010<sub>2</sub> switch (ss address mode) case 0: Rs 47: R12 = Rs 5A: R11 = Rs case 1: @Rs 47: A = Rs, R12 = Rs, word read start 04: wait read, R11 = Q case 2: (Rs)+ 47: A = Rs, Rs = Rs + 2, word read start 04: wait read, R11 = Q case 3: @(Rs)+ 47: A = Rs, word read start, Rs = Rs + 2 4E: wait read, R12 = Q, A = Q, word read start 04: wait read, R11 = Q case 4: -(Rs) 47: Rs = Rs - 2, A = Rs, word read start 04: wait read, R11 = Q case 5: @-(Rs) 47: Rs = Rs - 2, A = Rs, word read start 4E: wait read, R12 = Q A = Q, word read start 04: wait read, R11 = Q case 6: E(Rs) 47: A = PC, word read start, PC = PC + 2 46: wait read, R12 = Rs + Q, A = R12, word read start 04: wait read, R11 = Q case 7: @E(Rs) 47: A = PC, word read start, PC = PC + 2 46: wait read, R12 = Rs + Q, A = R12 word read start 4E: wait read, R12 = Q, A = Q, word read start 04: wait read, R11 = Q  if (dd address mode == 0) if (dd register == 7) </pre>



Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

Код инструкции	Мнемоника	NZVC	Описание
			<pre> 30: A = PC, R12 = PC+2, IR preload start word read start 59: Rd = R11, assign NVZ 11: next instruction else 30: A = PC, PC = PC + 2, IR preload start word read start 59: Rd = R11, assign NVZ, MCIR poll 01: next instruction else switch(dd address mode) case 0: Rd 45: user exception (000010<sub>8</sub>) case 1: @Rd 45: A = Rd, R12 = Rd, word rmw start case 2: (Rd)+ 45: A = Rd, Rd = Rd + 2, word rmw start 55: R12 = Rd - 2 case 3: @(Rd)+ 45: A = Rd, word read start, Rd = Rd + 2 4D: wait read, R12 = Q, A = Q, word rmw start case 4: -(Rd) 45: Rd = Rd - 2, A = Rd, word rmw start 55: R12 = Rd case 5: @-(Rd) 45: Rd = Rd - 2, A = Rd, word read start 4D: wait read, R12 = Q A = Q, word rmw start case 6: E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12, word rmw start case 7: @E(Rd) 45: A = PC, word read start, PC = PC + 2 4C: wait read, R12 = Rd + Q, A = R12 word rmw start 4D: wait read, R12 = Q, A = Rd, word rmw start 48: wait read, R12 = Q - R11, assign NZVC 19: Q = Q, write modified, data, assign NZVC 11: next instruction </pre>
17xxxx <sub>8</sub>		-	Недопустимая команда 01: user exception (000010 <sub>8</sub> )

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 10. Обнаруженные ошибки

Флаг переноса **C** трактуется инструкциями условного перехода как сброшенный после инструкций **MOVB xx, Rd** и **MFPS Rd** (только с указанными режимами адресации приемника) [7]. Ошибка локализована по микроадресу 0x51 (для 1801BM1A), на этом микроадресе выполняется знаковое расширение байта в регистр результата, при этом временное значение флага **C** в промежуточном регистре флагов устанавливается нулевым и может быть использовано следующей инструкцией условного перехода:

```
sec          ; установим флаг C
movb R1, R0  ; флаг C не должен измениться
bcc 1$       ; переход будет выполнен (неверно)
```

Значение флага **C** в реальном регистре **PSW** устанавливается всегда корректно, значение флага в промежуточном регистре влияет только на инструкции перехода.

```
sec          ; установим флаг C
movb R1, R0  ; флаг C не должен измениться
mfps R2      ; бит 0 в R2 будет единичным (верно)
bcc 1$       ; переход будет выполнен (неверно)
```

При выполнении любой инструкции (кроме **MOVB xx, Rd** и **MFPS Rd**) промежуточный регистр флагов перезаписывается из реального **PSW** и инструкции условного перехода начинают работать правильно

```
sec          ; установим флаг C
movb R1, R0  ; флаг C не должен измениться
nop          ;
bcc 1$       ; переход не будет выполнен (верно)
```

Микроадрес 0x51 используется инструкциями **MOVB xx, Rd** и **MFPS Rd**, а также инструкцией **SWAB Rd** (для выставления флагов **N** и **Z** по младшему байту результата), но инструкция **SWAB** предполагает сброс флага **C**, поэтому данная ошибка не проявляется.

Однокристалльный микропроцессор К1801ВМ1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801ВМ1@gmail.com	

## 11. Маркировка

Маркировка процессоров К1801ВМ1 в различные периоды их промышленного выпуска осуществлялась согласно начального ТУ и нескольких последующих изменений.

Начальное ТУ БК0.348.570-02ТУ-81	1.12.81 до 31.12.83	К1801ВМ1А - одна точка К1801ВМ1Б - две точки
Изменение ТУ №3 от 1.09.83г	1.01.84 до 31.12.89	К1801ВМ1А - одна точка К1801ВМ1Б - две точки К1801ВМ1В - три точки К1801ВМ1Г - четыре точки
Изменение ТУ №8 от 21.04.89г.	1.01.90	К1801ВМ1А - без точек К1801ВМ1Б - одна точка К1801ВМ1Г - две точки

Однокристалльный микропроцессор K1801BM1	Version: 1.2
Техническое описание	Date: 4-Jan-2020
Copyright by 1801BM1@gmail.com	

## 12. Ссылки

1. Сборное описание 1801BMx - <http://vak.ru/doku.php/proj/bk/1801vm1>
2. Статья о маркировке 1801BM1 - <http://sovietcpu.com/articles/69-label-1801>
3. Тема о реверс-инжиниринге 1801BM1 - <http://zx-pk.ru/showthread.php?t=23978>
4. Описание на 155la3.ru - <http://www.155la3.ru/k1801.htm>
5. Википедия - <https://ru.wikipedia.org/wiki/1801BMx>
6. ГОСТ 26765.51-86 Интерфейс МПИ - <http://rfgost.ru/gost/332002/download>
7. Статья об ошибке в 1801BM1- <https://m.habr.com/ru/post/471020/>