



КР1878ВЕ1

ОДНОКРИСТАЛЬНЫЙ МИКРОКОНТРОЛЛЕР

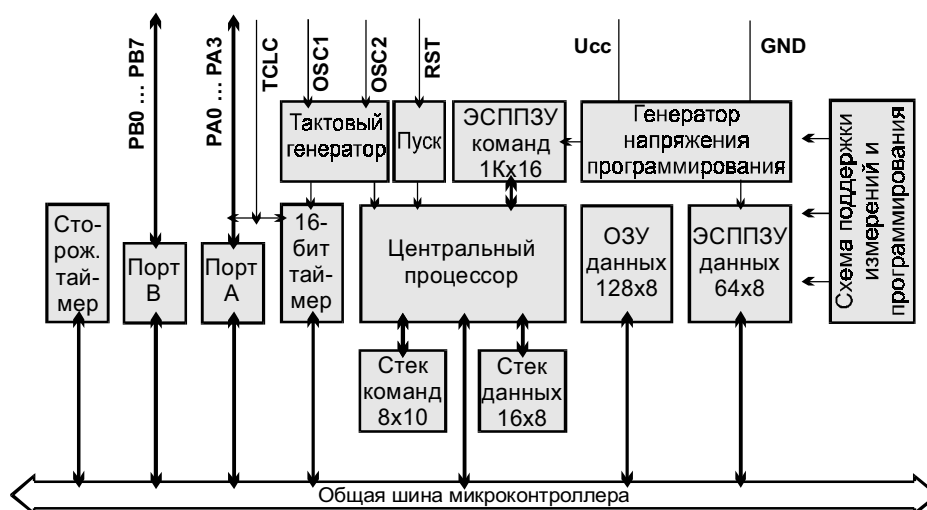
Микроконтроллер **КР1878ВЕ1 (An15E03)** является представителем семейства однокристальных микроконтроллеров разработки и изготовления российской полупроводниковой фирмы **“Ангстрем”**. Характерной особенностью данного семейства являются гарвардская **RISC** - архитектура, позволяющая выполнять любую из 52 команд за два такта частоты процессора, единая система команд для всего семейства с возможностью адресации до двух операндов, находящихся в памяти, малое время реакции на прерывание и сохранения контекста, широкий диапазон конфигураций внутренней памяти команд, памяти данных и внутренних периферийных устройств.

Микроконтроллер **КР1878ВЕ1** предназначен для использования в системах управления, работающих в масштабе реального времени. Микроконтроллер отличается малым количеством внешних выводов, низким током потребления, высокой производительностью, наличием энергонезависимой памяти данных и возможностью многократного перепрограммирования памяти команд.

ОСОБЕННОСТИ

- ✎ Память команд 1К x 16, реализованная на электрически стираемом ППЗУ (ЭС-ППЗУ)
- ✎ Память данных - ОЗУ 128 x 8
- ✎ Память данных - ЭСППЗУ 64 x 8
- ✎ Система команд - 52 команды
- ✎ Тактовая частота - 32 кГц .. 8 мГц
- ✎ Время выполнения любой команды - 250 нсек при частоте 8 мГц (2 такта)
- ✎ Прерываний - 7 (начальный пуск, системная ошибка, сторожевой таймер, порт А, порт Б, таймер, окончание записи в ЭСППЗУ)
- ✎ Время реакции на прерывание - 4 такта
- ✎ 12 линий ввода / вывода с индивидуальным управлением направлением и прерыванием от любой линии.
- ✎ Максимальный ток - 25 мА
- ✎ 16-разрядный таймер с 8-разрядным делителем счетной частоты
- ✎ Сторожевой таймер с автономным генератором
- ✎ Технология изготовления микросхемы - КМОП, два металла
- ✎ Ток потребления
 - < 2 мА при Vcc = 5 В и f = 5 мГц,
 - 50 мкА при Vcc = 5 В и f = 32 кГц,
 - < 1 мкА в режиме малого потребления STOP
- ✎ Корпус 18-выводной DIP типа 2104.18-В

СТРУКТУРНАЯ СХЕМА ИС



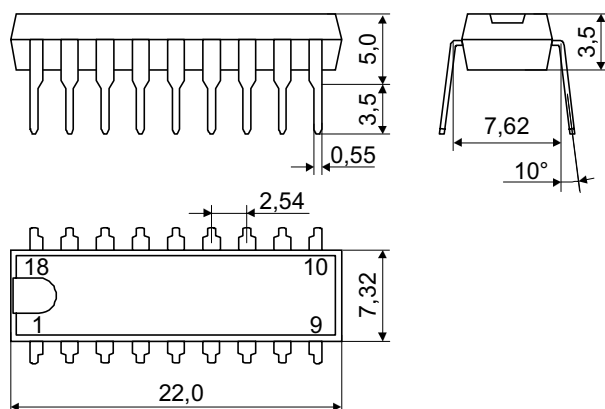


КОНСТРУКЦИЯ

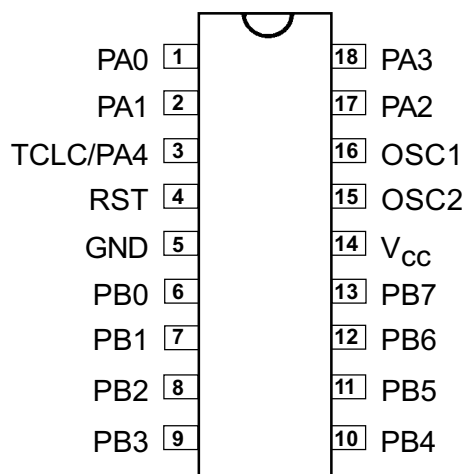
Микроконтроллер изготовлен по КМОП технологии и выпускается в пластмассовом 18-выводном корпусе DIP типа 2104.18-B. По заказу, при достаточном объеме партии микроконтроллер может быть изготовлен в ином конструктивном исполнении.

Микросхема предназначена для автоматизированной сборки аппаратуры и соответствует требованиям ГОСТ 20.39.405, группа IX, исполнение 2, а также для ручной сборки.

Корпус 21-4.18-B



Конфигурация выводов



ОПИСАНИЕ ВЫВОДОВ

Номер вывода	Символ	Назначение
1	PA0	Сигнал ввода-вывода порта А
2	PA1	Сигнал ввода-вывода порта А
3	TCLC/PA4	Внешняя частота таймера или сигнал ввода-вывода порта А
4	RST	Входной сигнал установки и программирования
5	GND	Общий вывод
6	PB0	Сигнал ввода-вывода порта Б
7	PB1	Сигнал ввода-вывода порта Б
8	PB2	Сигнал ввода-вывода порта Б
9	PB3	Сигнал ввода-вывода порта Б
10	PB4	Сигнал ввода-вывода порта Б
11	PB5	Сигнал ввода-вывода порта Б
12	PB6	Сигнал ввода-вывода порта Б
13	PB7	Сигнал ввода-вывода порта Б
14	V _{cc}	Напряжение питания
15	OSC1	Задающий вывод генератора тактовой частоты
16	OSC2	Задающий вывод генератора тактовой частоты
17	PA2	Сигнал ввода-вывода порта А
18	PA3	Сигнал ввода-вывода порта А



ЭЛЕКТРИЧЕСКИЕ ХАРАКТЕРИСТИКИ

 $T = (-40 \dots +85)^{\circ}\text{C}$

Параметр	Единица	Символ	Норма	
			Мин	Макс
Напряжение питания	В	U_{CC}	4,5	5,5
Напряжение сохранения данных в ОЗУ	В	U_{DB}	1,5	-
Выходное напряжение на выводах PA0-PA4, PB0-PB7, OSC2: - низкого уровня - высокого уровня	В	U_{OL} U_{OH}	- 3,8	0,6 -
Ток потребления: - статический - динамический: $f_i = 32 \text{ кГц}$ $f_i = 1,0 \text{ МГц}$ $f_i = 6,0 \text{ МГц}$	мА	I_{CC} I_{CC01} I_{CC02} I_{CC03}	-	1,0 1,4 3,0 8,0
Ток утечки на входе на выводах PA0-PA4, PB0-PB7, OSC1: - низкого уровня - высокого уровня	мкА	I_{LIL} I_{LIH}	-	5,0 5,0
Ток утечки на входе на выводе RST: - низкого уровня - высокого уровня	мкА	$I_{LIL(RST)}$ $I_{LIH(RST)}$	-	5,0 5,0
Емкость выходов	пФ	C_{IO}	-	50
Количество циклов перезаписи в ЭСППЗУ	-	N_{PR}	-	100 000
Время цикла записи в ЭСППЗУ	мс	T_{CYPR}	-	5,0

ПРЕДЕЛЬНО-ДОПУСТИМЫЕ РЕЖИМЫ ЭКСПЛУАТАЦИИ

Параметр	Единица	Символ	Норма			
			Предельно-допустимая		Предельная**	
			Мин	Макс	Мин	Макс
Напряжение питания	В	U_{CC}	4,5	5,5	4,0	6,0
Напряжение: - на любом входе (кроме RESET) - на входе RESET	В	U_{i1} U_{i2}	0 0	U_{CC} 13,2	-0,3 -0,3	$U_{CC}+0,3$ 13,5
Входное напряжение низкого уровня: - на выводах PA0 - PA4, PB0 - PB7 - на выводе RESET	В	U_{iL1} U_{iL2}	0 0	0,8* 0,7*	-0,3 -0,3	$0,16U_{CC}$ -
Входное напряжение высокого уровня: - на выводах PA0 - PA4, PB0 - PB7 - на выводе RESET	В	U_{iH1} U_{iH2}	$0,48U_{CC}$ 9*	U_{CC} 13,2	-	$U_{CC}+0,3$ 13,5

* - С учетом всех видов помех

** - Допускается работа микросхемы без гарантии параметров



ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ БЛОКОВ

Центральный процессор

Центральный процессор микроконтроллера **KP1878BE1** предназначен для выполнения арифметических и логических преобразований 8-разрядных операндов, расположенных в памяти микроконтроллера. Архитектура процессора характеризуется разделенной памятью команд и данных. Это позволяет совместить во времени процессы выборки команд и выборки операндов из памяти. Система команд процессора - симметричная, т.е. имеются двухоперандные команды, работающие одновременно с двумя операндами. Выборка операндов из памяти данных в случае двухоперандных команд производится одновременно по двум различным шинам - данных **SRC** операнда и данных **DST** операнда, с отдельными шинами адресов. Запись результата в память производится также по отдельной шине данных записи. Адрес записи совмещен с адресом чтения операнда **DST**.

Команды процессора имеют размерность 16 разрядов. Для обеспечения механизмов перехода к подпрограммам и прерываний программ в процессоре существует отдельный аппаратный стек глубиной в восемь адресов, где хранятся адреса возврата из подпрограмм и прерываний. В процессоре имеются служебные регистры для поддержки метода адресации операндов. Для сохранения значений этих регистров и слова состояния процессора при прерываниях программ и переходов к подпрограммам введен отдельный аппаратный стек данных глубиной в 16 байтов. Обращения к регистрам периферийных устройств микроконтроллера происходит по тем же шинам и точно так же как и к оперативной памяти.

Любая из команд процессора выполняется за одинаковый период времени. Одновременно на разных стадиях выполняются три команды процессора. Условные и безусловные переходы происходят по абсолютным адресам памяти команд, что позволяет устранить потерю времени при выполнении переходов.

Распределение адресного пространства

Регистры периферийных устройств: портов ввода/вывода, таймеров, ЭСППЗУ данных размещены в едином адресном пространстве вместе с оперативной памятью данных в фиксированной области внешних устройств, занимающих начальные 64 байта. Обращение к этим регистрам можно осуществлять любой из команд процессора точно также как и к байтам оперативной памяти.

Адрес ₁₆	Устройство	Адрес ₁₆	Устройство
0	Регистр состояния процессора	1B,1C	Свободный адрес
1	Рабочий регистр порта А	1D	Регистр управления сторожевого таймера
2	Рабочий регистр порта Б	1E÷37	Свободные адреса
3	Свободный адрес	38	Регистр управления ЭСППЗУ
4	Регистр управления таймера	39,3A	Регистр адреса ЭСППЗУ
5	Рабочий регистр таймера	3B÷3E	Свободные адреса
6÷18	Свободные адреса	3F	Регистр данных ЭСППЗУ
19	Регистр конфигурации порта А	40÷DF	Оперативная память
1A	Регистр конфигурации порта Б	C0÷FF	Свободные адреса

Система команд

Двухоперандные команды производят арифметические и логические операции над двумя операндами, адреса которых образуются из индексов, содержащихся в самой команде, и значений регистров адресов сегментов. Один из операндов не изменяет своего содержимого и обозначается далее сокращением **src** (источник). Результат операции помещается во второй операнд, обозначаемый **dst** (приемник).

Литерные команды производят арифметические и логические операции с операндом **dst**, и литерой **const** указанной в самой команде. Результат операции помещается в операнд, обозначаемый **dst** (приемник). Две команды **MOVL** и **CMPL** оперируют с полной восьмиразрядной литерой **const**, команды **ADDL** и **SUBL** с пятиразрядной литерой **sconst**, старшие три разряда заполняются нулями. Оставшиеся четыре команды в качестве операнда используют четырехразрядную (тетрадную) литеру **tconst**, местоположение которой в образуемой для литерного операнда **src** восьмиразрядной литере определяется разрядом местоположения тетрадной литеры - **p**. При 0-м разряде местоположения **p** тетрадная литера размещается в младшей тетраде полной литеры, а остальные разряды заполняются нулями. Установленный в единицу разряд местоположения **p** размещает тетрадную литеру в старшей тетраде, младшая тетрада полной литеры заполняется нулями.

Однооперандные команды производят арифметические или логические операции над операндом, адрес которого образуются из индекса, содержащегося в самой команде, и значения регистра адреса сегмента. Результат операции помещается в тот же операнд, обозначаемый **dst** (приемник).

Команда загрузки служебных регистров процессора предназначена для безусловной установки необходимых значений регистров адресов сегментов, для тех случаев, когда необходима адресация к новым адресам переменных или регистрам ввода/вывода, не охватываемых текущими сегментами. Также этой командой загружаются регистры косвенной адресации и устанавливаются режимы их работы. Служебные регистры могут сохраняться в памяти данных и восстанавливаться из нее при помощи команд **MFPR** и **MTPR**. Регистры адресов сегментов и регистры косвенной адресации доступны по номерам служебных регистров, указываемым в командах **LDR**, **MFPR** и **MTPR**. Для сохранения служебных регистров процессора при прерываниях и переходах к подпрограммам предназначен стек, размером 16x8. При прерываниях в стеке данных происходит автоматическое сохранение регистра состояния процессора. При выполнении команды возврата из прерывания сохраненное значение автоматически переписывается в регистр состояния. Для записи и чтения из стека значений служебных регистров имеются команды **PUSH** - записи в стек и **POP** - чтения из стека. Указатель глубины заполнения стека **DSP** (Data Stack Pointer) может меняться от 0 до 15. Команда **PUSH** увеличивает значение **DSP**, а команда **POP** - уменьшает. Ситуации, когда при значении **DSP**=15 подается команда **PUSH**, или **DSP**=0 подается команда **POP**, приводят к прерыванию по ошибке стеков. Команды работы с регистром состояний процессора **SST** и **CST** дают возможность установить или сбросить необходимые разряды в регистре состояний, указанные маской в теле этих команд.

Команды передачи управления производят передачу управления программе, находящейся по адресу, указанному в младших десяти разрядах команды перехода, либо безусловно, либо по одному из условий перехода. При переходе к подпрограмме в стек команд записывается адрес возврата. При возврате из подпрограммы или прерывания адрес, восстановленный из стека адресов



возврата (стека команд), записывается в счетчик команд. Указатель глубины заполнения стека команд **ISP** (Instruction Stack Pointer) может меняться от 0 до 7. Команда перехода к подпрограмме или прерывание увеличивает значение **ISP**, а команды возврата из подпрограммы или прерывания - уменьшают. Ситуации, когда при значении **ISP=7** подается команда перехода к подпрограмме или происходит прерывание, или при **ISP=0** подается команды **RTS** или **RTI**, приводят к прерыванию по ошибке стеков.

Специальные команды относятся к классу команд, не вписывающихся в предыдущие разделы, и предназначены в основном для управления определенными состояниями процессора.

Регистр состояния процессора содержит разряды, отражающие текущее состояние вычислительного процесса. Отдельные его разряды могут изменяться в процессе выполнения всех команд, некоторые - командами изменения регистра состояния. Установка разряда - это изменение состояния этого разряда в регистр состояния процессора всегда сохраняется в стеке данных, а при выполнении команды возврата из прерывания - восстанавливается из стека.

Представление отрицательных чисел при выполнении арифметических команд дается в дополнительном коде. Старший разряд байтового операнда несет информацию о знаке числа. У положительных чисел знаковый разряд равен нулю, а у отрицательных - единице. Наибольшее число $3F_{16} = 128_{10}$, а наименьшее - $80_{16} = -128_{10}$. Перенос из старшего разряда происходит при сложении двух отрицательных чисел и при сложении большего по модулю положительного числа с меньшим по модулю числом или равным по модулю отрицательным числом. Арифметическое переполнение происходит при сложении двух одинаковых по знаку чисел, если результат по модулю превышает 128_{10} .

Сокращения, принятые в таблице системы команд

s (src) – операнд источника;	a (address) – адрес команды;	c – устанавливаемое значение
d (dst) – операнд приемника;	n (number) – номер служебного разряда C в RS;	
c (const) – константа;	го регистра;	* – разряд PC меняется;
p (place) – местоположение	b (bit) – разряд регистра состояния (RS);	«-» – разряд PC не изменяется.
тетрады константы;		

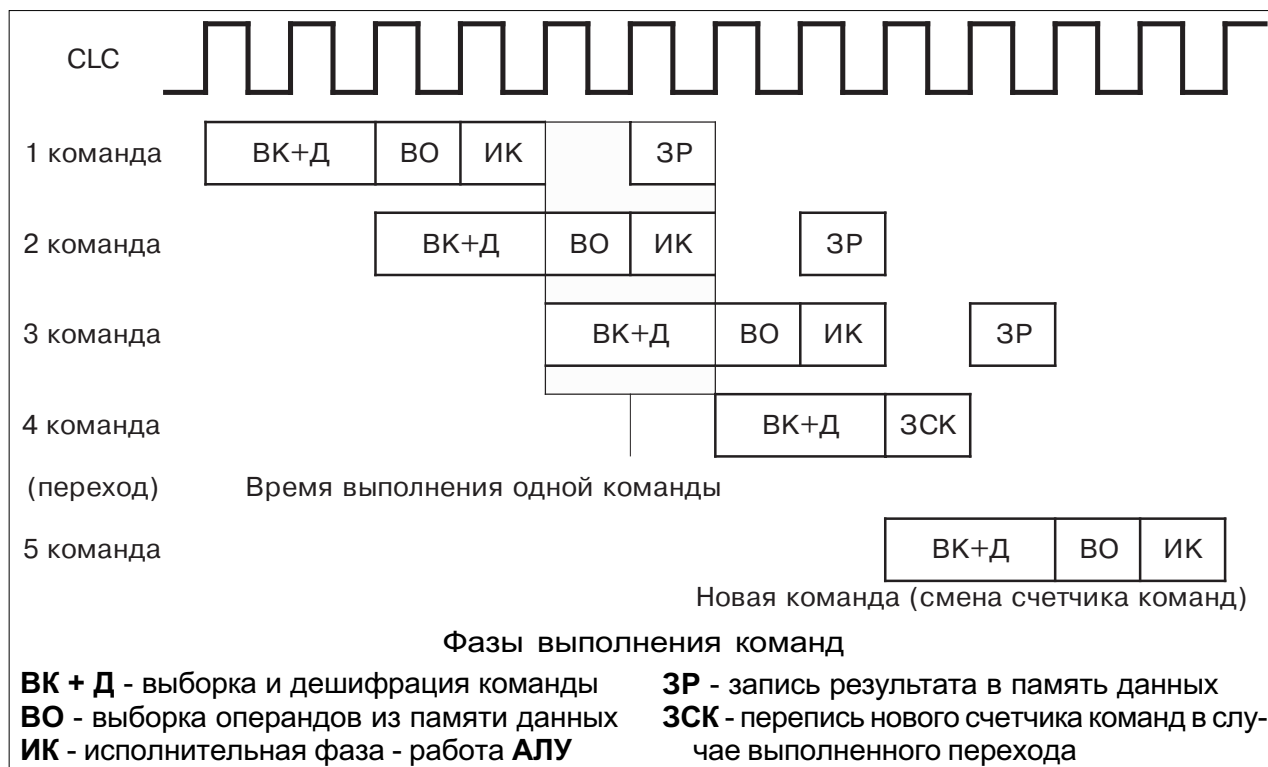
Временное представление потока команд

Для повышения производительности процессора производится совмещение в потоке команд отдельных операций, выполняемых внутренними устройствами микроконтроллера. На рисунке показан принцип совмещения по времени различных фаз выполнения команды на примере трех арифметических команд и одной команды перехода.

Выборка операндов для текущей команды и ее исполнение совмещаются с выборкой и дешифровкой последующей команды. Запись результата в память данных отложена на один такт для обеспечения нормальной загрузки информационных трактов памяти данных. Когда результат операции используется в качестве операнда для последующей команды, операнд берется из регистра результата на выходе **АЛУ**.

Таким образом, происходит совмещение по времени выборки и дешифровки (**n+1**)-й команды с выборкой операндов и исполнительной фазой **n**-й команды и записью результата (**n-1**)-й команды.

При обработке ситуаций, когда изменяется счетчик команд, после дешифровки команды перехода и соблюдения условия перехода происходит перепись нового счетчика команд из регистра команды в счетчик команд.



При обработке ситуаций, когда изменяется счетчик команд, после дешифрации команды перехода и соблюдения условия перехода происходит перепись нового счетчика команд из регистра команды в счетчик команд.

Блок ЭСППЗУ данных

Блок ЭСППЗУ данных включает в себя непосредственно сам накопитель данных размерностью в 64 байта и накопитель 4-разрядных контрольных разрядов, схему обнаружения и исправления однократной ошибки в байте данных, 8-разрядный регистр данных для хранения текущего байта, записываемого в накопитель или считываемого из накопителя ЭСППЗУ, и интерфейса для связи с шинами чтения и записи центрального процессора. Время хранения информации в ЭСППЗУ не менее 10 лет.

Обмен с блоком ЭСППЗУ данных осуществляется с помощью регистра управления, регистра адреса и регистра данных. В регистре адреса указывается адрес байта в накопителе. Доступ к необходимому байту в накопителе производится с помощью регистра данных. Регистр управления позволяет подавать команды чтения байта из накопителя в регистр данных и записи регистра данных в накопитель ЭСППЗУ. Окончание цикла записи может вызвать прерывание центрального процессора с вектором F_{16} . Для чтения байта из накопителя необходимо подать в регистр управления команду чтения байта, предварительно установив в регистре адреса требуемый адрес. По этой команде байт из накопителя переписывается в регистр данных. После этого можно считывать требуемый байт из регистра данных по его адресу. Для записи в накопитель необходимо записать в регистр данных требуемое значение байта, установить необходимый адрес в регистре адреса и подать команду записи в регистр управления. Предыдущее значение байта будет стерто и записано новое значение. Окончание процедуры записи проявится нулевым значением разряда записи в регистре управления или прерыванием, если был установлен разряд разрешения прерывания в регистре управления. Сигнал прерывания снимается любым обращением по чтению или записи к регистру управления.



Система команд

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

Продолжение ➤



Система команд. Продолжение

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

Порт ввода/вывода А

Порт ввода/вывода А предназначен для формирования на выводах **PA0...PA4** необходимых уровней напряжения, в случае, когда эти выводы используются в режиме выхода, и считывания значения уровней напряжения с этих выводов, как собственных, так и подаваемых извне, когда вывод **PA0..PA4** используется

в режиме входа.

Возможна инициация процедуры прерывания по изменению уровня напряжения на выводах **PA0..PA4**. Возникновение прерывания заключается в выработке сигнала прерывания, подающегося на блок прерываний центрального процессора с вектором равным **6**. Снятие сигнала прерывания происходит по чтению рабочего регистра порта А.

Задание режимов работы выводов порта и программирование прерываний от порта производится с помощью регистра конфигурации порта А. Изменение и регистрация уровней напряжения на выводе **PA0..PA4** производятся с помощью рабочего регистра порта А.

Порт ввода/вывода Б

Порт ввода/вывода Б предназначен для формирования на выводах **PB0..PB7** необходимых уровней напряжения, в случае, когда эти выводы используются в режиме выхода, и считывания значения уровней напряжения с этих выводов, как собственных, так и подаваемых извне, когда вывод **PB0..PB7** используется в режиме входа.

Возможна инициация процедуры прерывания по изменению уровня напряжения на выводах **PB0..PB7**. Возникновение прерывания заключается в выработке сигнала прерывания, подающегося на блок прерываний центрального процессора с вектором равным **7**. Снятие сигнала прерывания происходит по чтению рабочего регистра порта Б.

Задание режимов работы выводов порта и программирование прерываний от порта производится с помощью регистра конфигурации порта Б. Изменение и регистрация уровней напряжения на выводе **PB0..PB7** производятся с помощью рабочего регистра порта Б.

Интервальный таймер - счетчик внешних событий

Интервальный таймер - счетчик внешних событий (далее «таймер») предназначен для формирования определенных временных интервалов для привязки исполняемой программы к реальному времени или для подсчета количества внешних событий на входе внешней синхронизации таймера.

Программисту таймер доступен регистром управления и рабочим регистром. Регистр управления позволяет произвести запуск таймера, установить режим использования рабочего регистра, разрешить прерывания от таймера и обнаружить ошибочные ситуации при необработанном прерывании. Через рабочий регистр (в зависимости от состояния регистра управления) доступны регистр конфигурации таймера, счетный регистр или регистр интервала. Регистр конфигурации предназначен для программирования источника счетных импульсов и делителя частоты этих импульсов. 16-разрядный регистр интервала содержит значение, до которого будет производиться счет таймера, а 16-разрядный счетный регистр содержит текущее значение подсчитанных счетных импульсов.

Таймер производит подсчет импульсов, поступающих с выхода схемы программи-

руемого деления входной частоты, до значения, предварительно установленного в регистре интервала. Затем происходит сброс счетного регистра и подсчет повторяется. По завершению счетного интервала может быть выдан сигнал прерывания на центральный процессор с вектором **3**. Сигнал прерывания снимается по любому обращению к регистру управления таймером. Если сигнал прерывания не будет снят к моменту формирования следующего запроса на прерывание, в регистре управления устанавливаются разряды ошибки.

Для использования таймера в режиме формирования интервалов времени в регистр интервала записывается необходимое значение и, разрешив прерывание, запускается таймер на счет. В конце каждого интервала будет выдан сигнал прерывания и начнется счет нового интервала. Интервал можно задавать как в периодах тактовой частоты процессора **CLC**, так и в периодах внешней частоты, подаваемой на вход **TCLC** таймера. Для использования таймера в режиме счетчика внешних событий, в регистр интервала записывается нулевое значение интервала и по значению счетного регистра определяется количество внешних событий на входе **TCLC**.

Сторожевой таймер

Сторожевой таймер предназначен для отслеживания непредвиденных ситуаций, когда центральный процессор в результате каких либо сбоев находится в неопределенном состоянии.

Сторожевой таймер постоянно по включению питания производит подсчет импульсов тактовой частоты, поступающих через программируемый делитель частоты на вход 8-разрядного счетного регистра. В качестве тактовых счетных импульсов берется частота с внутреннего генератора, которая @ 30 кГц, или тактовая частота с выводов **OSC1** и **OSC2**. В первом случае счетный интервал сторожевого таймера будет зависеть от температуры окружающей среды, напряжения питания и значения коэффициента деления делителя частоты. Если за время полного пересчета счетного регистра не будет произведена запись разряда сброса счетчика, сторожевой таймер вырабатывает сигнал немаскируемого прерывания на центральный процессор с вектором **1**. Сигнал прерывания снимается по любой записи в регистр управления сторожевого таймера.

Программирование режимов работы сторожевого таймера, сброс счетного режима, установка коэффициента делителя частоты производятся записью в регистр управления сторожевого таймера, которая происходит только после записи по его адресу характерного кода AA_{16} . Запись в регистр управления обязательно должна производиться с нулевым значением старшего разряда записываемого байта. Если старший разряд записываемого байта равен единице, запись в регистр управления сторожевого таймера не будет произведена.

Возможно использование сторожевого таймера в качестве дополнительного интервального таймера, в котором формирование счетных интервалов производится с помощью тактовой частоты пересчета, коэффициента деления тактовой частоты и полного значения 8-разрядного счетного регистра. $T_{и} = T_{CLC} * K_{д} * 255,5$. Запуск сторожевого таймера на счет производится записью шестого разряда единицей в регистр управления сторожевого таймера.

СПЕЦИФИКА ПОДКЛЮЧЕНИЯ МИКРОКОНТРОЛЛЕРА

Включение питания и пуск микроконтроллера

Для управления пуском микроконтроллера имеется внешний вывод установки и программирования **RST**. Если напряжение на выводе **RST** имеет низкий уровень,

микроконтроллер находится в состоянии установки. После подачи на этот вывод высокого уровня напряжения (+5 В) происходит запуск на счет счетчика задержки начального пуска. Счетчик задержки начального пуска отсчитывает интервал времени равный 2^{10} импульсов частоты с выхода генератора тактовой частоты процессора. После завершения отсчета интервала задержки начального пуска центральный процессор начинает выполнять процедуру прерывания по начальному пуску с вектором прерывания равным 0. При этом, в отличие от обычных прерываний, записей в стек команд и стек данных не происходит. Счетчик задержки начального пуска может быть отключен сбросом соответствующих разрядов в строке конфигурации микроконтроллера, записываемой при программировании микроконтроллера. Необходимо снимать сигнал **RST** (выставлять высокий уровень на этот вывод) в тот момент, когда питание микроконтроллера U_{cc} имеет уровень не менее 3,5 В.

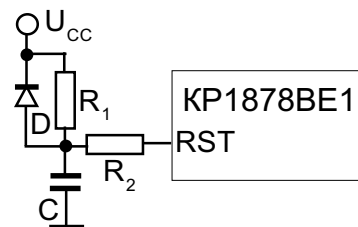
Для выработки сигнала **RST** может быть применена следующая схема подключения, где:

R1 – 40 кОм резистор, задающий высокий уровень на выводе **RST**.

R2 = 100 Ом – 1 кОм резистор, ограничивающий ток, протекающий через вывод **RST**.

C – конденсатор, задающий задержку формирования высокого уровня на выводе **RST**.

D – диод, обеспечивающий быстрый разряд конденсатора **C** при снятии напряжения питания.



Генерация тактовой частоты микроконтроллера

Микроконтроллер An15E03 имеет внутренний генератор тактовой частоты, который может работать в одном из четырех режимов:

- Генерация тактовой частоты с использованием внешнего кварцевого резонатора с частотами до 500 кГц;
- Генерация тактовой частоты с использованием внешнего кварцевого резонатора с частотами от 500 кГц до 8 МГц;
- Генерация тактовой частоты с использованием внешнего задающего резистора и емкости;
- Генерация частоты около 50 kHz с использованием внутреннего генератора;
- Режим трансляции внешней тактовой частоты.

Выбор режима работы генератора тактовой частоты осуществляется программированием соответствующих разрядов строки конфигурации микроконтроллера.

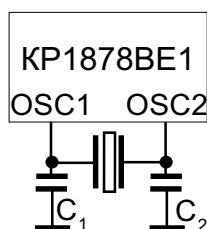


Схема подключения кварцевого резонатора

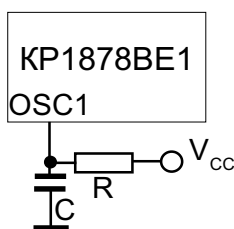


Схема подключения задающего резистора и конденсатора



Схема формирования тактовой частоты от внешнего генератора

СИСТЕМА ПРОЕКТИРОВАНИЯ И ОТЛАДКИ

Микроконтроллерное ядро **ТЕСЕЙ**, на основе которого разработан микроконтроллер KP1878BE1, включает комплекс программных и аппаратных средств для автоматизации проектирования, отладки и аттестации программ микроконтроллеров. Комплекс включает:

- компилятор Ассемблера микроконтроллеров **ТЕСЕЙ** - **TESSA 0.1**;
- пакет программ отладочной среды микроконтроллеров **ТЕСЕЙ**;
- аппаратный эмулятор микроконтроллеров **ТЕСЕЙ**.

TESSA 0.1 Компилятор Ассемблера микроконтроллеров ТЕСЕЙ

Язык **Ассемблер** и компилятор **TESSA 0.1** предназначены для разработки и отладки прикладных программ микроконтроллеров, построенных на основе ядра **ТЕСЕЙ**, в том числе МК **КБ5004BE1**, **KP1878BE1**. Отладка программ осуществляется либо на персональном компьютере в режиме эмуляции, либо на специальном аппаратном отладочном модуле. Для выполнения программирования и отладки необходимы IBM-совместимый персональный компьютер, пакет программ **TESSA 0.1** и программатор.

Отладочная среда микроконтроллеров ТЕСЕЙ

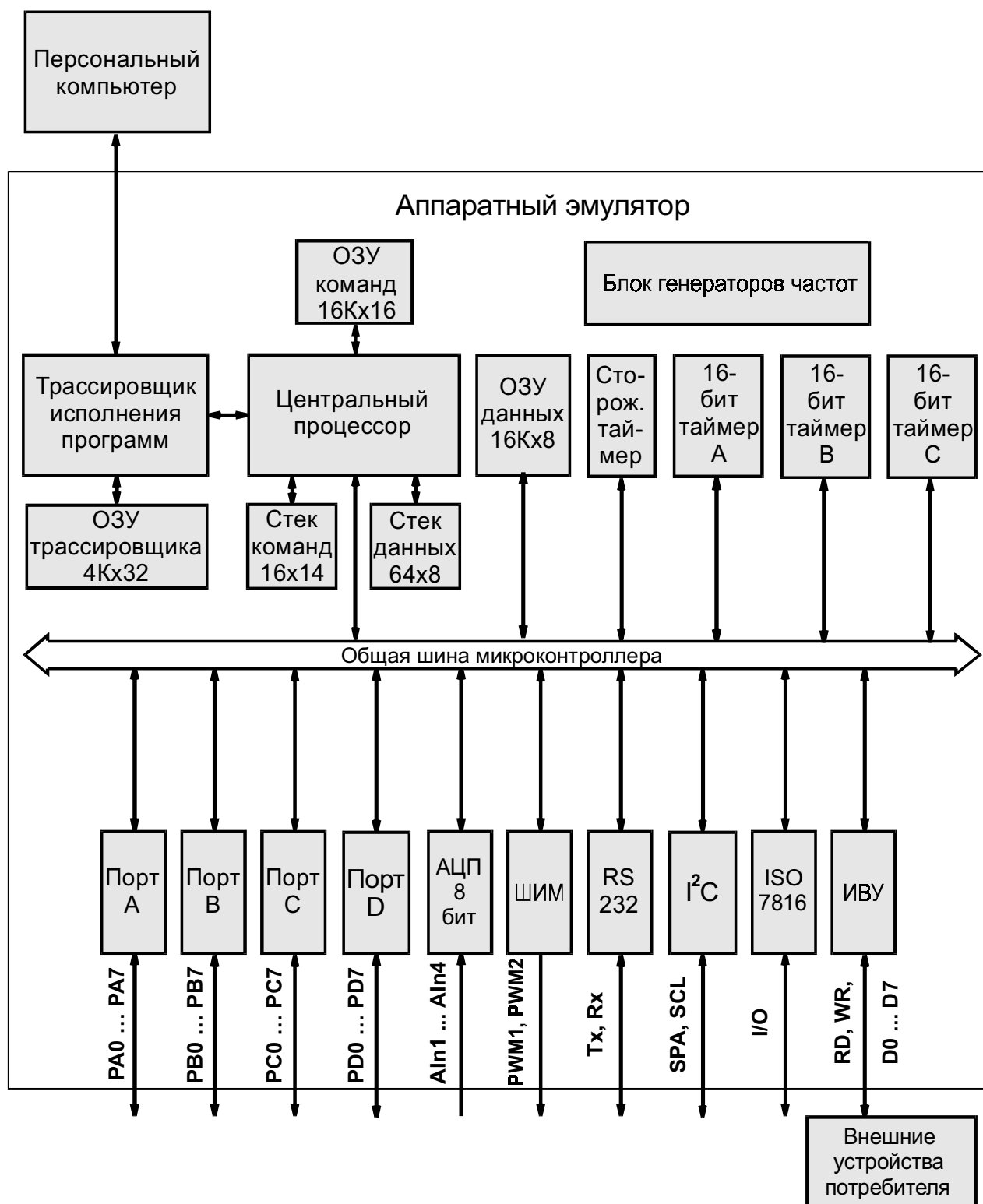
Отладочная среда предназначена для разработки и отладки при помощи языка Ассемблер и компилятора **TESSA 0.1** прикладных программ для KP1878BE1 и других микроконтроллеров, построенных на основе микроконтроллерного ядра **ТЕСЕЙ**. Отладка программ осуществляется либо на персональном компьютере в режиме эмуляции, либо на специальном аппаратном отладочном модуле. Для выполнения программирования и отладки необходимы IBM-совместимый персональный компьютер и пакет программ **TESSA 0.1**.

Аппаратный эмулятор микроконтроллеров ТЕСЕЙ

Аппаратный эмулятор (**АМ**) микроконтроллеров **ТЕСЕЙ** построен на основе специального отладочного кристалла, содержащего все основные модули микропроцессорного ядра **ТЕСЕЙ**. В результате отладка программ производится практически в резидентном режиме, что и обеспечивает максимально возможное качество отладки. **АМ** предназначен для отладки прикладных программ любых микроконтроллеров, построенных на основе ядра **ТЕСЕЙ**, в том числе микроконтроллеров **KP1878BE1**, **KP1878BE2**, **КБ5004BE1**, и др.

Программирование микроконтроллера

Запись разработанных программ в ЭСППЗУ команд микроконтроллера, строки конфигурации и начальных значений памяти ЭСППЗУ данных производится в режиме программирования. Переход микроконтроллера в режим программирования производится подачей на вывод **RST** высокого напряжения уровнем +12 В. Программирование микроконтроллера осуществляется с использованием выводов микроконтроллера **PB0**, **PB1** и **PB2**. Программирующее устройство (программатор) может записать и проконтролировать ЭСППЗУ команд, подавая определенные последовательности сигналов на эти выводы и анализируя состояния этих выводов. Следует отметить, что в начале программирования происходит полное обнуление всей памяти команд, и чтение ее содержимого можно осуществить только после записи в нее в пределах одного цикла программирования. Содержимое ЭСППЗУ данных и строки конфигурации останется прежним. Таким образом, полностью гарантируется конфиденциальность запрограммированной управляющей программы.

**Структура аппаратного эмулятора**

Строка конфигурации определяет режимы работы генератора тактовой частоты микроконтроллера и включение счетчика задержки начального пуска процессора. Следует отметить, что при программировании строки конфигурации каждый ее разряд необходимо повторять три раза подряд. Строка конфигурации при программировании рассматривается как 1025 слово ЭСППЗУ команд.

Формат строки конфигурации

Разряды	8	7	6	5	4	3	2	1	0
	RDE	RDE	RDE	OT2	OT2	OT2	OT1	OT1	OT1

RDE – разряд включения счетчика задержки начального пуска процессора. При **RDE=1**, счетчик задержки начального пуска процессора включен, при **RDE=0** – выключен.

OT1, OT2 – разряды выбора режима работы генератора тактовой частоты:

OT1=0, OT2=0 – режим генерации тактовой частоты с использованием внешнего задающего резистора и емкости;

OT1=1, OT2=0 – режим генерации тактовой частоты с использованием внешнего кварцевого резонатора с частотами от 500 кГц до 8 мГц;

OT1=(0), OT2=1 – режим подключения внутреннего генератора с частотой около 50 кГц;

OT1=(1), OT2=1 – режим генерации тактовой частоты с использованием внешнего кварцевого резонатора с частотами до 500 кГц, а также режим трансляции внешней тактовой частоты.



ПОКАЗАТЕЛИ УСТОЙЧИВОСТИ

Микросхема устойчива к механическим и климатическим воздействиям по ГОСТ 18 725 и ГОСТ 15150 (исполнение УХЛ категории 5.1), в том числе:

- линейным ускорениям $5\ 000\text{м/с}^2$ (500g) ,
- пониженной рабочей температуре среды -40°C ,
- повышенной рабочей температуре среды $+85^{\circ}\text{C}$,
- пониженной предельной температуре среды -60°C ,
- повышенной предельной температуре среды $+100^{\circ}\text{C}$,
- изменениям температуры среды от -60 до $+100^{\circ}\text{C}$.

ПОКАЗАТЕЛИ НАДЕЖНОСТИ

Наработка микросхемы на отказ:

- в полном диапазоне условий применения - 50 000 ч,
- в облегченном режиме (нормальные климатические условия и $U_{\text{cc}}=5\text{ В}$) - 60 000 ч.

Интенсивность отказов в течение наработки не более 1×10^{-6} 1/ч.

Гамма процентный срок сохраняемости 10 лет.

ГАРАНТИИ ИЗГОТОВИТЕЛЯ

Гарантии предприятия-изготовителя - по ГОСТ 18 725.

Гарантийный срок хранения 10 лет со дня изготовления.

Гарантийная наработка 50 000 ч. в пределах гарантийного срока хранения.



Обозначение микросхемы при заказе и в конструкторской документации и в конструкторской документации другой продукции:

Микросхема KP1878BE1 АДБК.431280.604ТУ.

Обозначение микросхемы при заказе и в конструкторской документации и в конструкторской документации по ГОСТ 20.39.405:

Микросхема KP1878BE1 АДБК.431280.604ТУ А.



Код ОКП: 63 3130 3161

Товарный штриховой код: 4601034507105