4-Х РАЗРЯДНЫЙ КМОП МИКРОПРОЦЕССОР КР1806ВЕ1

Процессор предназначен для использования в качестве контроллера в бытовой и промышленной аппаратуре. Области возможного применения микросхемы:

- бытовая техника средней сложности, игры;
- простые контроллеры промышленного оборудования;
- измерительные приборы и испытательная аппаратура;
- универсальный, дешевый и доступный процессор для радиоинженеров и радиолюбителей.

Технические характеристики:

- Напряжение питания - 4,0 -:- 6,5 вольт.
- Рабочий диапазон температур - -25 С -:- +70 С.
- Тактовая частота с реальной БИС ПЗУ - 0 -:- 1 МГц.
- Программы для процессора можно отлаживать в пошаговом режиме без использования логического анализатора.
- Максимальное быстродействие - 100 тыс. команд в секунду.
- Ток потребления при Fт = 30 кГц - менее 100 мкА.
- Ток потребления при Fт = 1 МГц - менее 1,0 мА.
- Разрядность АЛУ и аккумулятора - 4 разряда.
- Количество 4-х разрядных регистров - 4, байтовых - 4 шт.
- Объем ОЗУ на кристалле - 80 4-х разрядных слов.
- Максимальный объем ОЗУ - 8192 4-х разрядных слов.
- Максимальный объем ПЗУ - 8 кбайт.
- Таймер-счетчик событий - 8-ми разрядный.
- Прерывания от таймера и от входа микросхемы "TEST".
- Глубина стека - 3 уровня подпрограмм или прерываний.
- Количество двунаправленных портов ввода/вывода, размещенных на кристалле - четыре 4-х разрядных порта.
- Возможно расширение до 12 внешних портов.

Конструктивные данные:

- Корпус микросхемы - 42-х выводный пластмассовый QIP (тип - 2204.42) аналогичен корпусам БИС КР1801ВМ1, КР1801ВП1 и КР1806ВП1.
- Минимальная система - 3 -:- 4 микросхемы :
    1. стандартное ПЗУ или СППЗУ - 563РЕ1 или 573РФ2;
    2. 8-ми разрядный регистр адреса - К561ТМ3, К561ИР9 по 2 шт. или К561ИР6;
    3. микропроцессор КР1806ВЕ1.
- Микросхема легко сопрягается с интегральными схемами серий 561, 573, 537, 555 и т.д.

Особенности архитектуры процессора.

Процессор, по сравнению с другими 4-х разрядными микроконтроллерами (TMS1000, COP400, 1814ВЕ1, 1820ВЕ1), имеет существенно более развитую систему команд :
- все команды переходов и переходов на подпрограммы, в том числе и косвенные, имеют по 7 условий выполнения;
- команды косвенного обращения к ОЗУ могут адресоваться по 2-м парам регистров;
- в процессоре реализована команда косвенного чтения ПЗУ, что дает возможность операции умножения производить табличным способом и просто организовать дешифрацию и перекодировку информации;

Программное обеспечение.

В настоящее время разработку систем на процессоре можно производить с помощью программ "Интерпретатор"и "Ассемблер", работающих в Turbo-подобной среде с встроенными редактором и программой помощи "HELP" на микро-ЭВМ типа IBM PC и (без редактора) на ДВК-3,4.

Процессор не является копией какой-либо зарубежной БИС.
Цена БИС составляет 60 руб. на 09 1992 г.

Адрес для контактов и консультаций:
103482 г.Москва К-482 НИИТТ и завод "Ангстрем".
Телефон: г.Москва 532-86-22, 532-88-10, местный: 23-76.
Разработчик: Сафронов Аркадий Вячеславович.

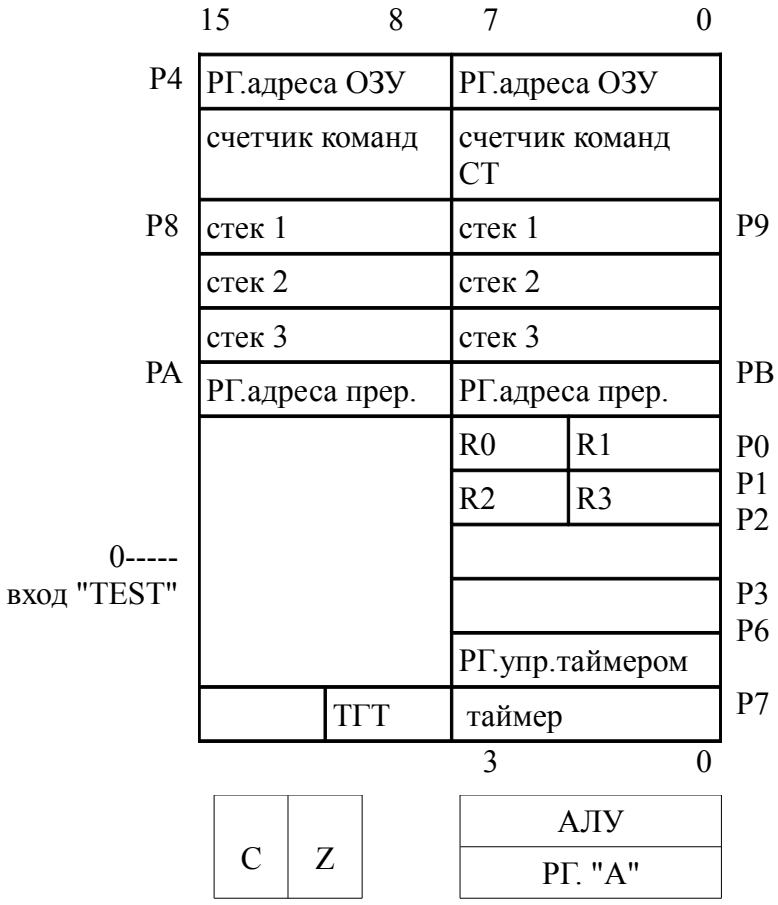
1 ОПИСАНИЕ МИКРОПРОЦЕССОРА КР1806ВЕ1

(АДБК.431280.110 ТУ КОД ОКП: 63 3125 8261)

- Технология - КМОП с 3-х мкм поликремниевыми затворами.
- Напряжение питания - 4.0 -:- 6,5 вольт.
- Максимальная тактовая частота (при Епит = 5 В) - 1 МГц.
- Ток потребления при Fт = 0 Гц - около 100 мкА.
- Ток потребления при Fт = 1 МГц - около 1 mA.
- Диапазон рабочих температур -25 -:- +70 С.
- В схеме процессора используются квазидинамические элементы, поэтому при отладке программ он может работать на сколь угодно низкой тактовой частоте при условии, что длительность импульса тактового генератора не будет превышать 25 мкс.
- Минимальная тактовая частота меандра - 20 кГц.
- На кристалле имеется схема кварцевого генератора, работающая при подключении внешнего кварцевого резонатора.
- Максимальное быстродействие - 100 тыс. команд в секунду.
- Разрядность АЛУ и аккумулятора - 4 разряда.
- Количество 4-х разрядных регистров - 4 шт (R0 -:- R3), к ним можно обращаться и как к парам регистров (Р0,Р1).
- Количество байтовых регистров - 4 шт (Р0 -:- Р3).
- Объем ОЗУ на кристалле - 80 4-х разрядных слов.
- Максимальный объем внешнего ОЗУ - 8 кслов по 4 разряда.
- Максимальный объем ПЗУ - 8 кбайт.
- Таймер-счетчик событий - 8 разрядов, на входе которого установлен программируемый делитель частоты.
- Возможны маскируемые прерывания от таймера и от входа " TEST ".
- Глубина аппаратного стека - 3 уровня подпрограмм или прерываний.
- Количество размещенных на кристалле двунаправленных портов ввода/вывода - 4 по 4 разряда в каждом.
- Максимальное количество внешних портов ввода/вывода- 12.
- Все выходы процессора могут быть подключены к емкостной нагрузке величиной до 50 пФ.
- Внутренние порты, выдающие логическую "1", могут управлять резистивной нагрузкой величиной свыше 2 кОм (вытекающий ток до 2 mA).
- Из кристалла выведена совмещенная 8-ми разрядная магистраль адресов/данных, поэтому младшие 8 адресов необходимо сохранять во внешнем адресном регистре.
- Процессор может быть переведен в режим "ОСТАНОВ" в котором ток потребления - около 150 мкА. Выход из этого режима происходит при прерывании процессора.
- Количество элементов в БИС - 10081.

- Корпус QIP - типа 2204.42 - пластмассовый 42-х выводный , аналогичный корпусам микросхем КР1801ВМ1, КР1801ВП1 и КР1806ВП1.
- Размеры корпуса с выводами: 26 * 25 * 6 мм.

2 АРХИТЕКТУРА МИКРОПРОЦЕССОРА КР1806ВЕ1



Счетчик команд - СК.
C - триггер переноса АЛУ.
Z - флаг процессора - равен "1" только в том случае, если аккумулятор (РГ. "А") равен нулю.
ТГТ - триггер таймера.
R0 -:- R3 - 4-х разрядные регистры.
P0 -:- PB - 8-ми разрядные пары регистров.
РГ.адреса прер. - регистр адреса прерывания.

В пары регистров P4 и P6 можно только записывать информацию.

3 Система команд

3.1 Команды условных переходов

КОД ОПЕРАЦИИ	БАЙТЫ	ЦИКЛЫ
0000 0XXX [0X/H] переход длинный	3	3
0000 1XXX [0X/H] переход длинный на подпрограмму	3	3
0001 0XXX [1X/H] переход короткий	2	2
0001 1XXX [1X/H] переход короткий на подпрограмму	2	2
0010 0XXX [2X/H] переход косвенный по P0	1	1
0010 1XXX [2X/H] переход косв. на подпрогр. по P0	1	1

/ XXX / - УСЛОВИЕ ПЕРЕХОДА

---

000 [0,8/H] безусловный
001 [1,9/H] при "C"=0
010 [2,A/H] при "C"=1
011 [3,B/H] при "Z"=0
100 [4,C/H] при "Z"=1
101 [5,D/H] при "ТГТ"=1
110 [6,E/H] при "TEST"=1

(/H - шестнадцатеричная система счисления.)

Цикл состоит из 4-х тактов тактовой частоты процессора.

Максимально-возможный объем ПЗУ, составляющий 64 кбайта, разбит на 256 страниц по 256 байт в каждой, причем переход с одной страницы на другую происходит только при выполнении команды длинного перехода.

Младшая половина счетчика команд - счетчик, а старшая - регистр, поэтому для переходов в пределах страницы лучше использовать команды косвенного и короткого переходов, а для переходов на другие страницы (даже в том случае, если не изменяется естественный порядок следования команд) - обязательна команда длинного перехода.

Условия перехода "ТГТ"=1 и "TEST"=1 проверяются в первом цикле исполнения команды.

ПРИМЕР 1:

```
адрес     команда      ## условие: "C"=1 ##
 658F    0000 1010 [0A/H] - команда дл. перехода
 6590    0111 1001 [79/H]  на подпрогрограмму
 6591    1011 0001 [B1/H]  по адресу 79B1 при "C"=1.
 79B1    .... ....
          адрес 6592 посылается в стек.
```

ПРИМЕР 2:

```
адрес     команда      ## условие: "Z"=1 ##
 52E6    0000 0011 [03/H] - команда длинного пере-
```

```
52E7    1001 0100 [94/H]   хода по адресу 94F5
52E8    1111 0101 [F5/H]   при "Z"=0.
52E9    .... ....
```

ПРИМЕР 3:

```
адрес     команда      ## условие: "ТГТ"=1 ##
176E    0001 1101 [1D/H] - команда кор. перехода
176F    0010 0001 [21/H]   на подпрогрограмму
1721    .... ....          по адресу 21 при "ТГТ"=1.
                           адрес 1770 посылается в стек.
```

ПРИМЕР 4:

```
адрес     команда      ## условие: "С"=1 ##
488F    0001 0001 [11/H] - команда короткого перехода
4890    0011 1001 [39/H]   по адресу 39 при "С"=0.
4891    .... ....
```

ПРИМЕР 5:

```
адрес     команда      ## условие: P0=50 ##
4192    0010 1000 [28/H] - команда безусловного
4150    .... ....          косвенного перехода на подпрогрогр. по адресу,
                           находящемуся в P0.
                           адрес 4193 посылается в стек.
```

3.2 Безадресные команды

КОД ОПЕРАЦИИ ВСЕ КОМАНДЫ - 1 БАЙТ, 1 ЦИКЛ.

```
0011 0100 [34/H]  возврат из подпрограммы
0011 1000 [38/H]  РГ адреса прерывания (РГАП) -> СК
0100 0100 [44/H]  нет операции (НОП)
0100 0101 [45/H]  режим "ОСТАНОВ" процессора
0100 0110 [46/H]  режим аппаратного стека
0100 0111 [47/H]  режим программного стека
0101 0100 [54/H]  запуск таймера
0101 0101 [55/H]  останов таймера
0101 0110 [56/H]  установка в "0" триггера "С"
0101 0111 [57/H]  установка в "1" триггера "С"
0110 0100 [64/H]  режим "АР"
0110 0101 [65/H]  режим "+С"
0110 0110 [66/H]  режим "ЛОГ"
0111 0100 [74/H]  циклический сдвиг аккумулятора вправо
0111 0101 [75/H]  установка в "0" триггера таймера
0111 0110 [76/H]  (НЕ"С") -> "С"; инверсия триггера "С"
0111 0111 [77/H]  (НЕ А) -> А;    инверсия аккумулятора
```

При выполнении команды "длинного косвенного перехода" - 38/H происходит переход по

адресу, находящемуся в регистре адреса прерывания(РГАП) РА и РВ (стек в этой команде не используется).

Команда сдвига 74/Н производит циклический сдвиг вправо только аккумулятора (триггер "С" не изменяется).

Режим "ОСТАНОВ" позволяет сократить ток потребления БИС на любой частоте до величины менее 150 мкА. Выход из этого режима происходит при прерывании или инициализации процессора.

Результаты операций, выполняемых процессором команд сложения и вычитания, зависят от того в каком из 3-х режимов процессор находится:

- арифметический -"АР";
- с учетом переноса -"+С";
- логический -"ЛОГ".

3.3 Команды работы с регистрами и ОЗУ

КОД ОПЕРАЦИИ

0011 XXXX [3X/H] A -> X
чтение из аккумулятора

	реж."АР"	реж."+С"	реж."ЛОГ"
0100 XXXX [4X/H]	A-X (C,Z)	A-X (C,Z)	A(И)X (Z)
	сравнение с аккумулятором		
0101 XXXX [5X/H]	A-X -> A,C*	A-X+C -> A,C*	A(И)X -> A
	вычитание из аккумулятора		
0110 XXXX [6X/H]	A+X -> A,C*	A+X+C -> A,C*	A(M2)X -> A
	сложение с аккумулятором		
0111 XXXX [7X/H]	X -> A	X -> A	X -> A
	запись в аккумулятор		

/ XXXX /	СПОСОБ АДРЕСАЦИИ	БАЙТЫ	ЦИКЛЫ
0000 [0/H]	регистр R0	1	1
0001 [1/H]	регистр R1	1	1
0010 [2/H]	регистр R2	1	1
0011 [3/H]	регистр R3	1	1
1001 [9/H]	ОЗУ косвенная адресация по Р0, "0" страница***	1	2
1010 [A/H]	ОЗУ косвенная адресация по Р1, "0" страница ***	1	2
1011 [B/H]	ОЗУ косв. адресация по Р0**	1	2
1100 [C/H]	ОЗУ косв. адресация по Р1**	1	2
1101 [D/H]	ОЗУ "0" страница ***	2	3
1110 [E/H]	ОЗУ длинная адресация	3	4

1111 [F/H] ОЗУ короткая адресация** 2 3

* - возможно изменение триггера "С".
** - старший байт (адрес страницы ОЗУ) - не изменяется и выдается на внешние выводы процессора.
*** - старший байт (адрес страницы ОЗУ) - сохраняется, а на внешние выводы БИС выдается адрес нулевой страницы.

"М2" и "И" - поразрядные логические операции: сложение по модулю 2 и логическое "И" (триггер переноса "С" не изменяется).

Флаг процессора "С": при сложении хранит информацию о наличии переноса в последней команде сложения, а при вычитании - об отсутствии заема, поэтому перед операцией вычитания необходимо триггер "С" устанавливать в"1".

Флаг процессора "Z" равен "1" только в том случае, если аккумулятор равен нулю.

ПРИМЕР:

адрес	команда
	## условия: P0 = 06,P1 = 9C ##
9442	0011 1110 [3E/H] - команда записи
9443	0100 1101 [4D/H] аккумулятора
9444	0010 1111 [2F/H] в ячейку ОЗУ
4D2F	обращение к ОЗУ по адресу 4D2F.
9445	0110 1111 [6F/H] - сложение аккумулятора
9446	1000 1010 [8A/H] с ячейкой ОЗУ
4D8A	обращение к ОЗУ по адресу 4D8A.
9447	0101 1101 [5D/H] - вычитание из аккумуля-
9448	0010 0101 [25/H] тора ячейки ОЗУ
0025	обращение к ОЗУ с адресом 0025.
9449	0101 1011 [5B/H] - вычитание из аккумуля-
4D06	обращение к ОЗУ тора ячейки ОЗУ (4D06).
944A	0111 1001 [79/H] - запись в аккумулятор
0006	обращение к ОЗУ из внешнего порта (6).
944B	0100 1100 [4C/H] - сравнение аккумулятора
4D9C	обращение к ОЗУ и ячейки ОЗУ (4D9C).

Вычитание в процессоре в "арифметическом" режиме производится так: происходит сложение аккумулятора с инверсным значением операнда и единицей в младшем разряде.
Команды сложения и вычитания в режиме "+С" производятся с учетом триггера "С": к младшему разряду результата прибавляется "1", если "С" равно "1".

Команды сравнения выполняются следующим образом :
содержимое регистра, ячейки ОЗУ или операнд в команде вычитается из аккумулятора, но результат не записывается в аккумулятор, а происходит только установка флажков "С" и "Z",которые фиксируют: был ли перенос при вычитании и был ли равен нулю результат вычитания. Результат сравнения сохраняется на время выполнения следующей команды, после чего восстанавливаются прежние состояния триггеров "С" и "Z", установленные в результате исполнения предыдущих команд сложения или вычитания. Результат выполнения команды сравнения:

"Z"= 1 - при равенстве содержимого аккумулятора и операнда;
"C"= 1 - при условии, что аккумулятор больше или равен операнду.

Максимально-возможный объем ОЗУ, составляющий 64 Кслова, разбит на 256 страниц по 256 4-х разрядных слов в каждой, причем переход с одной страницы на другую происходит только при выполнении любой команды обращения к ОЗУ с длинной адресацией или при непосредственной записи в регистр адреса страницы ОЗУ - 8-ми разрядный регистр Р4.

Адресация к "0" странице полезна при необходимости быстрого доступа к двум массивам переменных : команды с коротким адресом работают с какой - либо страницей ОЗУ, а перемежающиеся с ними команды обращения к "0" странице, не изменяя в регистре Р4 номер рабочей страницы ОЗУ, аппаратно обнуляют старшие адреса микропроцессора и позволяют работать также и в "0" странице ОЗУ.

Порты ввода/вывода процессора эквивалентны ячейкам ОЗУ с адресами 0 -:- F/H . Операции с портами ввода/вывода производятся только при выполнении команд обращения к ОЗУ со способом адресации по "0" странице. Внутренние порты ввода/вывода (4 порта по 4 разряда в каждом) размещены на кристалле БИС, а для использования внешних портов требуется подключение дополнительных микросхем.

3.4 Распределение адресного пространства ОЗУ

страница 0	страница 1
0000 внутренний порт 0	0100 не используется
0001 внутренний порт 1	0101 не используется
0002 внутренний порт 2	0102 не используется
0003 внутренний порт 3	0103 не используется
0004	0104
.. } внешние порты	.. } не используются
000F	010F
0010	0110
.. } не используются	.. } не используются
0017	0117
0018	0118
.. } внутреннее ОЗУ	.. } внутреннее ОЗУ
003F	013F
0040	0140
.. } внешнее ОЗУ	.. } внешнее ОЗУ
00FF	01FF

При адресации ко всем другим страницам ОЗУ происходит обращение только к внешнему ОЗУ.

В последнем цикле исполнения команды обращения к внешнему ОЗУ появляется "0" на выходе "CRAM", а к внешним портам ввода/вывода - "1" на выходе " CIO ".

Выход микросхемы "WR" становится равным "0" в цикле обращения к ОЗУ при выполнении команд записи из процессора во внешнее ОЗУ или во внешние порты ввода/вывода.

Для увеличения количества портов ввода и вывода удобно использовать ИС типа 561ЛН1 и 561ТМ3, 561ИР9 соответственно. Дешифрация внешних портов производится при участии сигналов:"CIO", "WR" и 4-х младших адресов, получаемых с адресного регистра. При большом количестве внешних портов ввода можно применять ИС типа 561КП1: 2 микросхемы позволяют построить систему с 16-ю входами.

3.5 Команды операций с парами регистров

КОД ОПЕРАЦИИ	БАЙТЫ	ЦИКЛЫ
1000 RRRR [8X/H] P0 -> PX	1	1
запись из пары P0 в любую пару регистров;		
1001 RRRR [9X/H] PX -> P0	1	1
запись из любой пары регистров в пару P0;		
1010 0RRR [A0-:-A7/H] P1 -> PX [X=0-:-7];	1	1
запись из пары P1 в любую пару регистров (P0 -:- P7);		
1011 0RRR [B0-:-B7/H] PX -> P1 [X=0-:-7];	1	1
запись из любой пары регистров (P0 -:- P7) в пару P1;		
1010 1RRR [A8-:-AF/H] D8 -> PX [X=0-:-7];	2	2
запись второго байта команды в любую пару регистров(P0 -:- P7);		
1011 1RRR [B8-:-BF/H] ПЗУ(P0) -> PX [X=0-:-7];	1	2
запись байта из ячейки ПЗУ с адресом - содержимым пары P0 в любую пару регистров (P0 -:- P7);		

RRRR и RRR - адреса пар регистров PX 0-:-F и 0-:-7.
D8 - константа(2-й байт команды).

Команды 80/H,90/H,94/H,96/H,A1/H,B1/H,B4/H и B6/H использовать запрещается.

Команда записи байта из ячейки ПЗУ с адресом - содержимым пары регистров P0 в любую пару регистров PX выполняется только в той странице ПЗУ, в которой находится код операции команды. Команда удобна для реализации программным способом дешифраторов и табличных функций и ее можно использовать для написания, например, программы умножения по таблице.

ПРИМЕР:

адрес	команда
8400	1010 1000 [A8/H] - команда записи байта

```
8401     0111 0011 [73/H]   "73"  в пару P0.
8402     1011 1011 [BB/H] - команда  записи байта
8473      обращение к ПЗУ   из ПЗУ по адресу 8473
8403     .... ....          в пару P3.
```

3.6 Команды работы с константами

ВСЕ КОМАНДЫ - 1 БАЙТ, 1 ЦИКЛ.

КОД ОПЕРАЦИИ	реж."АР"	реж."+С"	реж."ЛОГ"
1100 DDDD [CX/H]	A-D (C,Z)	A-D (C,Z)	A(И)D (Z)
сравнение константы с аккумулятором			
1101 DDDD [DX/H]	A-D -> A,C*	A-D -> A**	A(И)D -> A
вычитание константы из аккумулятора			
1110 DDDD [EX/H]	A+D -> A,C*	A+D -> A**	A(M2)D -> A
сложение константы с аккумулятором			
1110 0110 [E6/H]	ДКА	ДКА	A(M2)6 -> A !
десятичная коррекция аккумулятора.			
1111 DDDD [FX/H]	D -> A	D -> A	D -> A
запись константы в аккумулятор			

DDDD - константа D в команде - 4 бита.
* - возможно изменение триггера "С".
** - триггер "С" не изменяется.

Команда десятичной коррекции аккумулятора нужна при работе с десятичными числами (это числа, состоящие только из цифр от 0 до 9). При сложении операндов в диапазоне 0 -:- 9 сумма может получиться недесятичной, но, после использования команды ДКА, в результате получится десятичное число. Выполняется эта команда так:

если A > 9 или "С" = 1, то: A + 6 -> A и 1 -> "С".

Так как код операции "E6" занят командой ДКА, то просто прибавить к аккумулятору константу "6" нельзя - прибавляйте 2 раза по "3".

В арифметическом режиме выполнение команд сложения и вычитания эквивалентно работе процессора с ОЗУ и регистрами, а в режиме "+С" не изменяется триггер "С", что используется при операциях с многоразрядными числами, так как это позволяет считать количество обработанных 4-х разрядных чисел и вычислять адреса для косвенной адресации, сохраняя значение триггера "С".

3.7 Организация вложенных программ

В режиме аппаратного стека используются 3 уровня стека:
при переходе на подпрограмму: адрес из ПЗУ -> СК -> стек1 -> стек2 -> стек3;
при прерывании: РГАП -> СК -> стек1 -> стек2 -> стек3;
при возврате из прерывания и подпрограммы: стек3 -> стек2 -> стек1 -> СК.

В режиме программного стека используется 1 уровень стека:
при переходе на подпрограмму: адрес из ПЗУ -> СК -> стек1;
при прерывании: РГАП -> СК -> стек1;
при возврате из прерывания и подпрограммы: стек1 -> СК.

Режим программного стека можно использовать для организации стека произвольной глубины, но ответственность за сохранение адресов возврата через пару Р0 и аккумулятор в ОЗУ в этом случае несет программист.

3.8 Функциональное назначение разрядов регистра управления таймером (р6)

7-й разряд- "1": разрешение прерывания от | по
(старш.) триггера таймера "ТГТ"; |
6-й разряд- "1": разрешение прерывания от |"ИЛИ"
входа "TEST"; |
5-й разряд- "1": подключение входа предв. | по
делителя частоты к Fosc/4; |
4-й разряд- "1": подключение входа предв. |"И"
делителя частоты к входу |
"TEST"; |
3-й разряд-"0": 1. вход "TEST" поступает на схему прерывания без запоминания в RS триггере;
2. обнуление RS триггера;
"1": вход "TEST" поступает на схему прерывания с запоминанием в RS триггере;

2-й,1-й и 0-й разряды: управляют коэффициентом деления предварительного делителя частоты включенного на входе таймера:

Код	2-й разряд	0	1	0	1	0	1	0	1
	1-й разряд	0	0	1	1	0	0	1	1
	0-й разряд	0	0	0	0	1	1	1	1
Коэфф. деления предварительного делителя частоты		1	2	3	6	5	10	15	30

Таймер - вычитающий 8-ми разрядный счетчик.

При P6.4 = 1 и P6.5 = 1 можно измерять длительность импульса на входе "TEST".

Триггер таймера "ТГТ" переключается в "1" в следующих случаях:

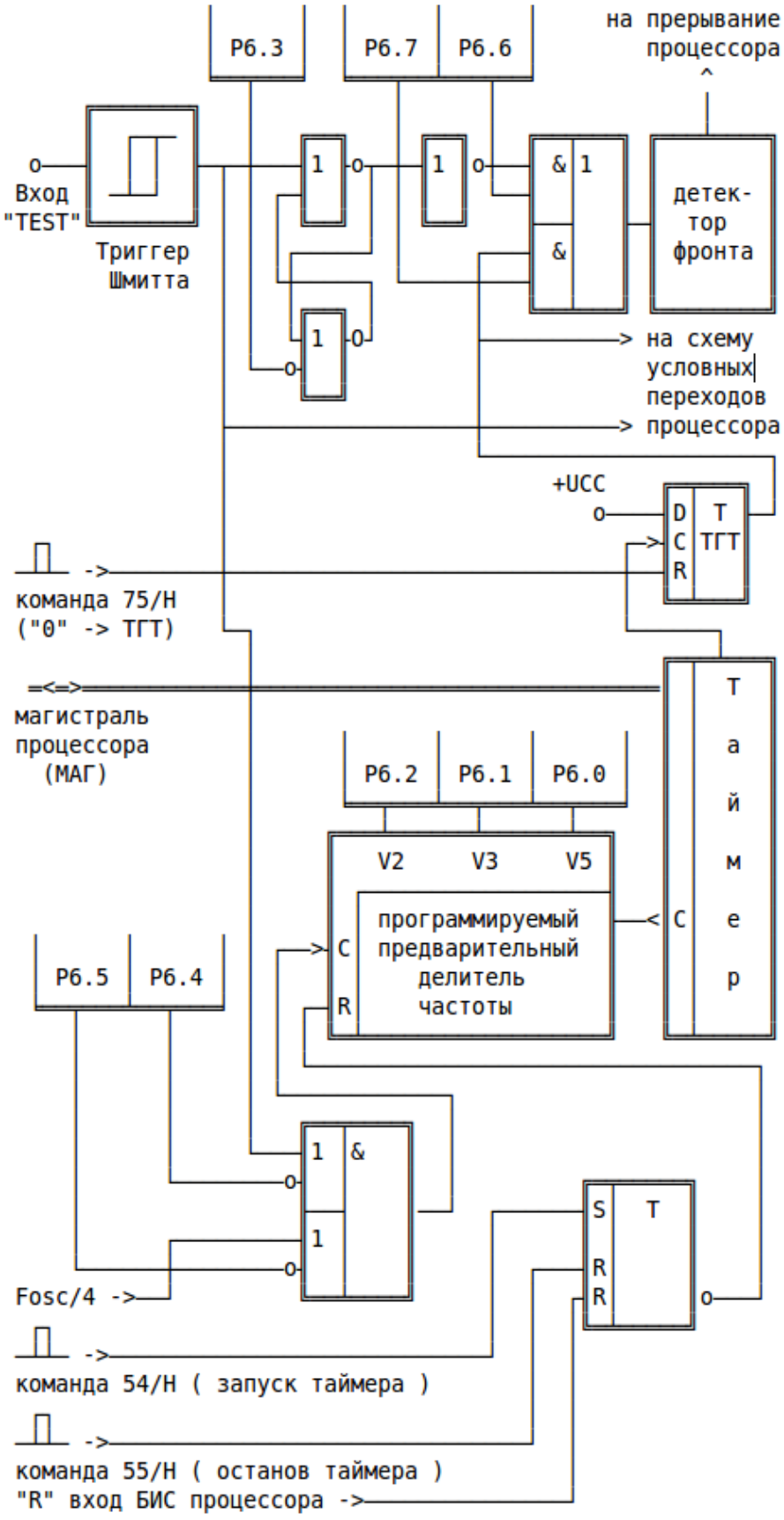
1) содержание счетчика-таймера в результате вычитания становится равным "00000000";
2) при записи в таймер любого числа, если до этого содержание счетчика-таймера было равно "00000000".

Для исключения ошибочной установки триггера таймера в состояние "1" запись в счетчик-таймер необходимо обязательно производить в такой последовательности:

1) записать в счетчик-таймер нужное значение;
2) сбросить триггер "ТГТ";
3) разрешить прерывание по "ТГТ".

Нельзя записывать в работающий таймер.

4 Схема управления таймером и прерыванием процессора



5 Прерывание процессора

Прерывание процессора - переход по адресу, содержащемуся в парах регистров РА и РВ - регистре адреса прерывания (при этом адрес команды, следующей за последней выполненной, записывается в стек) происходит в том случае, если в первом цикле выполнения очередной команды складываются условия для прерывания процессора. Если же эти условия возникают при исполнении команды сравнения, то оно задерживается до окончания выполнения следующей за ней команды (обычно это команда условного перехода). Поэтому минимальная длительность импульса прерывания на входе " TEST " равна 7 циклам процессора или 28 тактам Fosc - это время выполнения команды сравнения и следующей за ней команды условного перехода с наибольшей длительностью.

Выход из режима "ОСТАНОВ" происходит при прерывании или при инициализации процессора.

Часто возникает такая ошибка при работе с таймером и при использовании прерываний: как только триггер таймера перебросился в "1" - происходит прерывание,а в начале программы обработки прерываний стоит команда сброса в "0" триггера таймера [75/Hex]. Но, если коэффициент деления ПДЧ не 1 или 2, то в таймере еще остается "0" и потому триггер таймера снова перебрасывается в "1", и происходит незапланированное прерывание и чаще всего ошибка в работе программы. Для того, чтобы система работала правильно надо сбрасывать триггер таймера или через количество циклов процессора больше коэффициента деления ПДЧ или записать в таймер ненулевое значение.

6 Конструктивное исполнение БИС

Корпус - пластмассовый типа 2204.42, аналогичен корпусам БИС КР1801ВМ1, КР1801ВП1 и КР1806ВП1.

Размеры корпуса без выводов: 26 * 18 * 4 мм.

Размеры корпуса с выводами: 26 * 25 * 6 мм;

Выводы размещены с шагом 2,5 мм в 4 ряда с внутренней колеей 20 мм и внешней - 25 мм.

Рисунок фрагмента печатной платы приведен в схеме отладочной системы (см. стр 21).

При заказах более 10000 шт. в год возможно корпусирование БИС в миниатюрных пластмассовых микрокорпусах с размерами около 20 * 20 * 2,5 мм (включая выводы) с четырехсторонним размещением планарных выводов с шагом 1-:-1,25 мм.

В микросхеме КР1806ВЕ1 выведено только 13 адресных шин, поэтому максимальный объем ПЗУ и ОЗУ составляет соответсвенно 8 Кбайт и 8192 4-х разрядных слова.

Объем ПЗУ может достигать 64 кбайт, ОЗУ - 64 кслов в том случае, если вывести все 16 разрядов адреса. При необходимости увеличения количества адресов до 16 придется сократить количество внутренних портов или использовать корпус с большим числом выводов.

В будущем при заказах более нескольких десятков тысяч микросхем в год на кристалле можно разместить ПЗУ объемом до 1 -:- 2 кбайт и заказные схемы ввода-вывода.

В 1994 году разрабатывается вариант БИС КА1806ВЕ2 размещаемый в корпусе с 64 выводами на 4 стороны размером примерно 20 * 20 мм с внутренним ПЗУ объемом 2 кбайта и 8 4-х разрядных порта. Микросхема может работать как с внутренним масочным ПЗУ, так и с внешним ПЗУ.

7 Цоколевка микропроцессора в 42 выводном корпусе QUIP

1	CA	выход	строб регистра адреса (активный уровень — "0", активный фронт - "0-1");
2	TEST	вход	вход прерывания и вход таймера
3	R	вход	вход инициализации процессо ра (активный уровень- "1");
4	OSC1	вход	выводы для подключения
5	OSC2	выход	кварцевого резонатора;
6	PI23	вход/выход	порт 2 разряд 3;
7	PI22	вход/выход	порт 2 разряд 2;
8	PI21	вход/выход	порт 2 разряд 1;
9	CROM	выход	"выбор кристалла" ПЗУ (активный уровень- "0");
10	WR	выход	запись/чтение; (запись в ОЗУ, В/В->WR="0");
11	PI20	вход/выход	порт 2 разряд 0;
12	PI33	вход/выход	порт 3 разряд 3;
13	PI32	вход/выход	порт 3 разряд 2;
14	PI31	вход/выход	порт 3 разряд 1;
15	PI30	вход/выход	порт 3 разряд 0;
16	PI13	вход/выход	порт 1 разряд 3;
17	PI12	вход/выход	порт 1 разряд 2;
18	PI11	вход/выход	порт 1 разряд 1;
19	PI10	вход/выход	порт 1 разряд 0;

20	PI03	вход/выход	порт 0 разряд 3;	
21	PI02	вход/выход	порт 0 разряд 2;	
22	PI01	вход/выход	порт 0 разряд 1;	
23	PI00	вход/выход	порт 0 разряд 0;	
24	GND		общий вывод;	
25	CIO	выход	"выбор кристалла" внешней системы ввода/вывода (активный уровень- "1", активный фронт - "1-0");	
26	CRAM	выход	"выбор кристалла" ОЗУ (активный уровень- "0");	
27	AD0	вход/выход	Магистраль команд, данных, адресов;	(0 разряд)
28	AD1	вход/выход		(1 разряд)
29	AD2	вход/выход		(2 разряд)
30	AD3	вход/выход		(3 разряд)
31	AD4	вход/выход		(4 разряд)
32	AD5	вход/выход		(5 разряд)
33	AD6	вход/выход		(6 разряд)
34	AD7	вход/выход		(7 разряд)
35	A8	выход	адрес (8 разряд);	
36	A9	выход	адрес (9 разряд);	
37	A10	выход	адрес (10 разряд);	
39	A11	выход	адрес (11 разряд);	
40	A12	выход	адрес (12 разряд);	
42	UCC		источник питания +UCC;	

8 Описание выводов микросхемы процессора

Все выводы БИС КР1806ВЕ1 защищены от воздействия статического электричества в такой-же степени, как и другие КМОП ИС, например, серии К561 или К537, то есть не

требуется никаких особых мер для предохранения БИС от поражения статическим электричеством. Но разумная осторожность не помешает.

Для уменьшения помех по источнику питания до величины по крайней мере 5% от напряжения питания БИС к выводам "Ucc" и "GND" микросхемы рекомендуется подключить керамический конденсатор емкостью не менее 68 нФ. Особенно внимательно к помехам нужно относиться при использовании в системе ТТЛ ИС и мощных ключей.

Микросхемы из всех партий функционировали на частоте 125 кГц при снижении напряжения питания до 2,2 В, но использовать их в этом режиме не рекомендуется до набора достаточной статистики. Данное сообщение сделано только с целью показать достаточность запасов по минимальному напряжению питания БИС.

8.1 Входы "R" и "TEST"

Если на вход инициализации микросхемы "R" в течение не менее 4 тактов Fosc подать напряжение соответствующее логической "1", то происходит :

1) обнуление всего программного счетчика;

2) обнуление регистра управления таймером;

3) выключение таймера.

4) обнуление внутреннего "указателя стека".

5) обнуление внутренних портов.

И потому для полной однозначности работы процессора в начале программы необходимо программно определить:

! ! ! 1) режим работы стека - программный или аппаратный;

! ! ! 2) режим работы АЛУ "АР","+С" или "ЛОГ";

3) обнулить триггер "ТГТ".

На входах "TEST" И "R" для повышения помехоустойчивости установлены триггеры Шмитта с гистерезисом около 1 вольта. Неиспользуемый вход "TEST" обязательно подключать к отрицательному выводу источника напряжения.

В реальных системах при наличии помех по питанию и.т.д. возможен такой эффект: процессор воспринимает, например, адрес в команде перехода как код операции и тогда может произойти "зацикливание" или "зависание" процессора и прекращение функционирования микропроцессорной системы.

В обслуживаемых системах восстановление функционирования производится с помощью кнопки "Reset", подающей напряжение Ucc на вход "R" БИС. Но иногда такой вариант невозможен и тогда используется другой способ известный из литературы под названием "Watchdog" (сторожевая собака): в цикле программы происходит периодическое обнуление интегратора, подключенного к входу "R" и, в случае "зацикливания" или "зависания", происходит заполнение интегратора до порогового значения и, следовательно, обнуление процессора. В качестве интегратора можно использовать счетчик на ИС или RC-цепочку. Пример, описанный в файле pri.rus, и схема из подсправочника \RIS показывают вариант схемы "Watchdog" на RC-интеграторе.

8.2 Вход "OSC1" и выход"OSC2"

Для функционирования процессора необходимо подавать на БИС импульсы тактовой частоты. Возможно испольъзование внешнего генератора тактовых импульсов, в этом случае выход внешнего кварцевого или RC - генератора подключается к выводу " OSC1 " микросхемы КР1806ВЕ1, длительность фронтов этих импульсов не должна превышать 1 мксек.

В схеме процессора используются квазидинамические элементы, поэтому при отладке программ он может работать на сколь угодно низкой тактовой частоте при условии, что длительность "1" импульса тактового генератора не будет превышать 25 мкс.

Минимальная тактовая частота меандра - 20 кГц.

Для получения тактовой частоты можно использовать в качестве активного элемента кварцевого генератора инвертор, подключенный к выводам микросхемы "OSC1" (вход) и "OSC2"(выход).

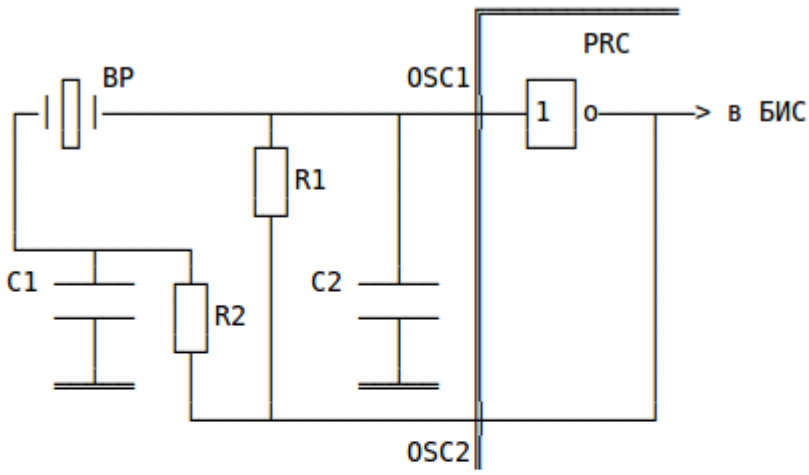


СХЕМА ПОДКЛЮЧЕНИЯ КВАРЦЕВОГО РЕЗОНАТОРА К БИС КР1806ВЕ1

таблица примерных номиналов элементов схемы

F кГц	R1 МОм	R2 МОм	C1 пФ	C2 пФ
32,768 1000	1 -:- 5 0,5 -:- 1	0,15 -:- 0,33 0	56 56	33 33

Точные номиналы элементов схемы кварцевого генератора уточняются экспериментально для каждого типа используемого резонатора. При невысоких требованиях к стабильности тактовой частоты можно заменить кварцевый резонатор дросселем (при индуктивности 1 мГн и 2-х конденсаторах емкостью 47 пФ тактовая частота Fosc получается порядка 1 МГц).

К выходу "OSC2" может быть подключена внешняя резистивная нагрузка более 500 кОм и емкостная до 10 пФ дополнительно к нагрузке кварцевого резонатора и проводников платы минимальной длины (до 20 мм).

Если используется внутренний кварцевый генератор процессора, то время подачи логической "1" на вход "R" необходимо увеличить на время выхода на режим кварцевого резонатора (для резонаторов с частотой 32768 Гц это время может составлять несколько сотен миллисекунд).

8.3 Все выходы и входы-выходы магистрали

Из кристалла выведена совмещенная 8-ми разрядная магистраль адресов-данных, поэтому младшие 8 адресов необходимо сохранять во внешнем адресном регистре, запись в который производится по сигналу "CA".

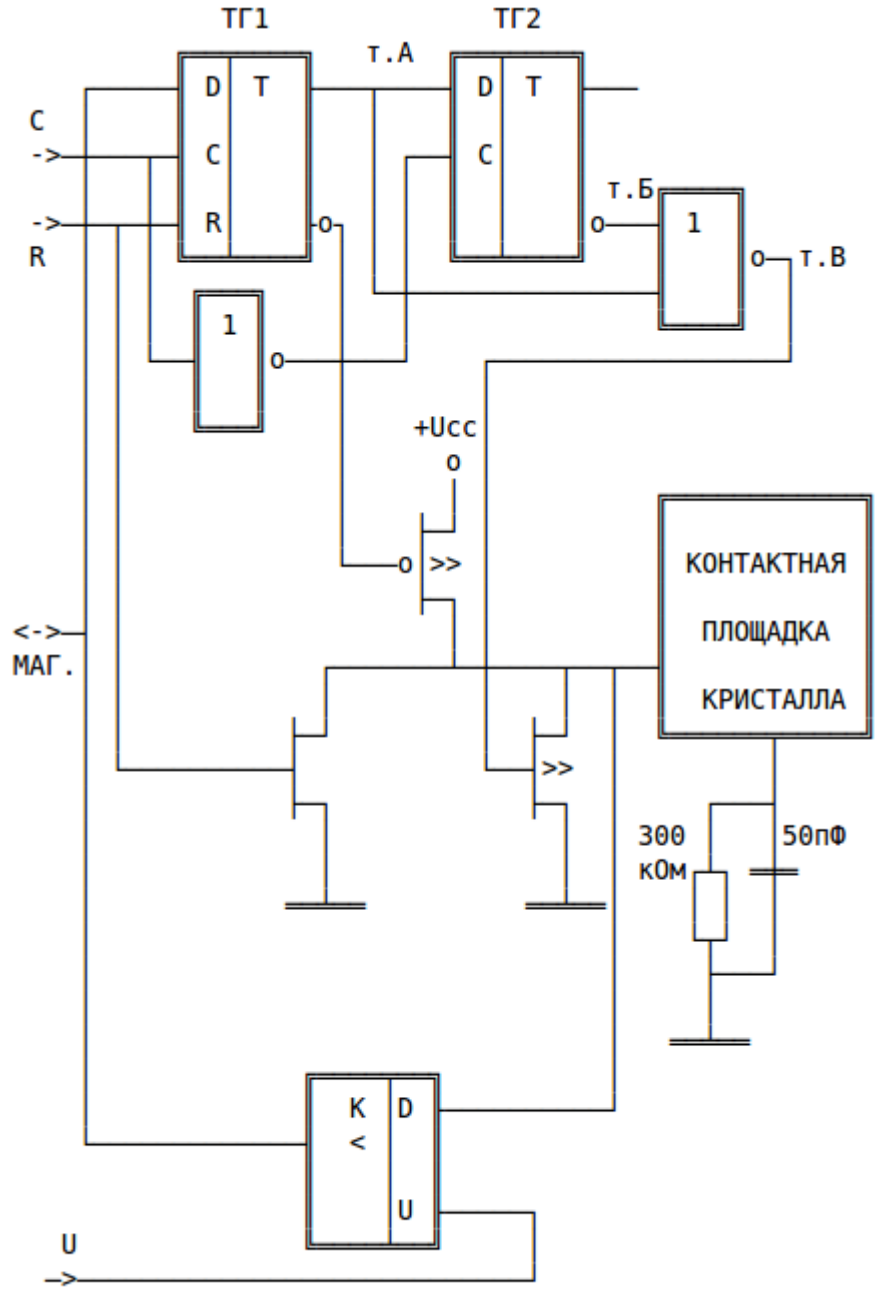
Младшие адреса внешних ПЗУ и ОЗУ подключаются к выходам адресного регистра, старшие - к выходам A8-:-A12 процессора. Выборка кристаллов ПЗУ и ОЗУ производится сигналами "CROM" и "CRAM" соответственно, а определение режима ОЗУ - запись или чтение определяется сигналом на выходе "WR" БИС КР1806ВЕ1. 4-х разрядные внешние ОЗУ и порты ввода и вывода подключаются к младшим разрядам 8-ми разрядной магистрали адресов-данных AD0 -:- AD3 микросхемы.

Сигналы с выходов БИС можно непосредственно подавать на входы NMOП и ТТЛ ИС, но, при использовании ТТЛ микросхем с входным током более 0,7 мА, рекомендуется применять усилители тока на NPN транзисторах.

Микросхема КР1806ВЕ1 с КМОП ИС серий 561, 564 и 537 стыкуется без каких-либо особенностей, но и в этом случае нужно помнить, что при подключении комбинационных КМОП ИС (не триггеры и регистры) к выводам магистрали, которые могут находиться в высокоимпедансном состоянии, возможно увеличение тока потребления этими микросхемами от источника питания. Выводы магистрали процессора КР1806ВЕ1 схемотехнически защищены от этого эффекта и "подтяга" магистрали не требуется.

Однако, при использовании БИС совместно с NMOП и ТТЛ интегральными схемами необходимо учитывать, что логическая "1" на выходе этих ИС существенно ниже напряжения питания и может достигать величины 2,4 вольт, а порог переключения входных инверторов большинства КМОП ИС (в том числе и КР1806ВЕ1) находится в диапазоне от трети до двух третей напряжения питания, и поэтому, если сигналы с выходов NMOП и ТТЛ ИС поступают на входы любых КМОП ИС, то необходимо установить между выходами NMOП и ТТЛ ИС и положительным выводом источника питания резисторы соротивлением 10 -:- 27 кОм (СППЗУ типа 573РФ2 и 2716 некоторых изготовителей установки этих резисторов не требуют). По этой же причине микросхема на частоте, близкой к максимальной, может не работать из-за недостаточности времени "подтягивания" выходов ПЗУ резисторами до порога. Система будет функционировать быстрее, если на вход "CE" БИС СППЗУ подать проинвертированный сигнал "CA",а на "OE"-"CROM".

8.4 Входы-выходы внутренних портов



Примечания:

1. Элемент "К" - повторитель с третьим состоянием для передачи информации из порта во внутреннюю магистраль процессора (МАГ).
2. 2. знаком ">>" обозначены мощные транзисторы.

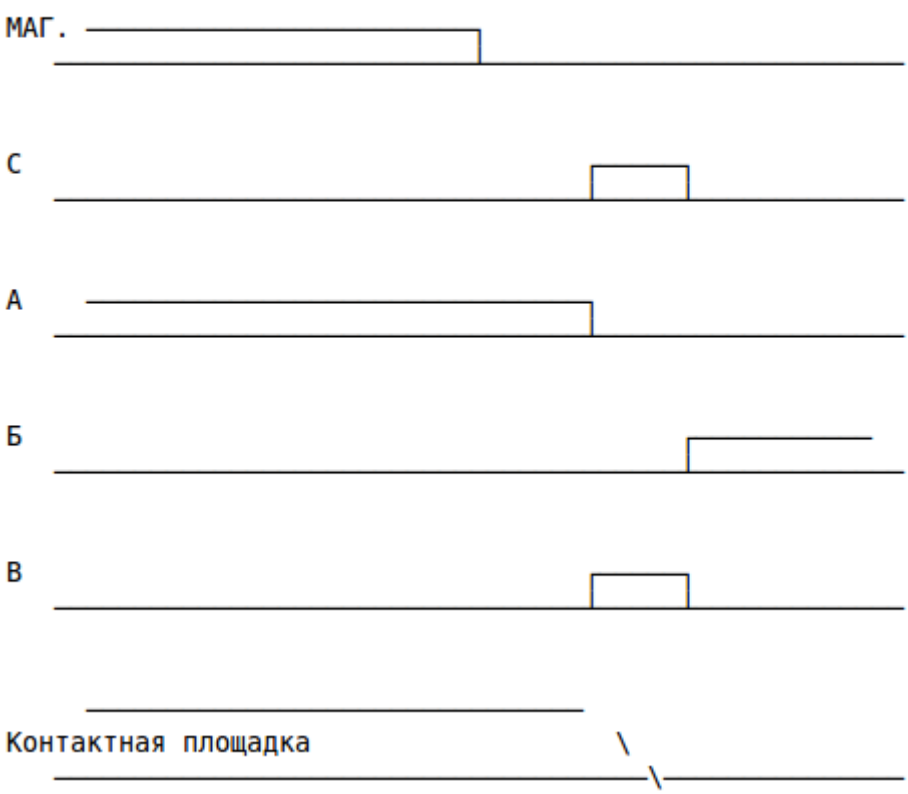
Внутренние порты ввода/вывода двунаправленные и каждый разряд порта активно выдает только "1", а "0" удерживается внешним резистором 300 кОм. В этом случае вре мя

обнуления нагрузочной емкости составило бы :

3 * 300 кОм * 50 пФ = 50 мксек.

Это время не может удовлетворить пользователей и для устранения этого эффекта введены триггер ТГ2, вентиль ИЛИ-НЕ и NMOП транзистор, который включается на короткое время и быстро разряжает емкость нагрузки.

ВРЕМЕННАЯ ДИАГРАММА РАБОТЫ ВНУТРЕННЕГО ПОРТА



Итак особенности использования внутренних портов:

1. Любой разряд внутреннего порта может быть запрограммирован на ввод информации, если в этот разряд записать логический "0".

2. При использовании разряда внутреннего порта в режиме вывода информации необходимо между выводом порта микропроцессора и общим выводом установить резистор номиналом не более 300 кОм(если порт управляет ключом выполненном на NPN транзисторе, то достаточно поставить базовый резистор).

3. Выходы могут быть подключены к емкостной нагрузке величиной до 200 пФ.

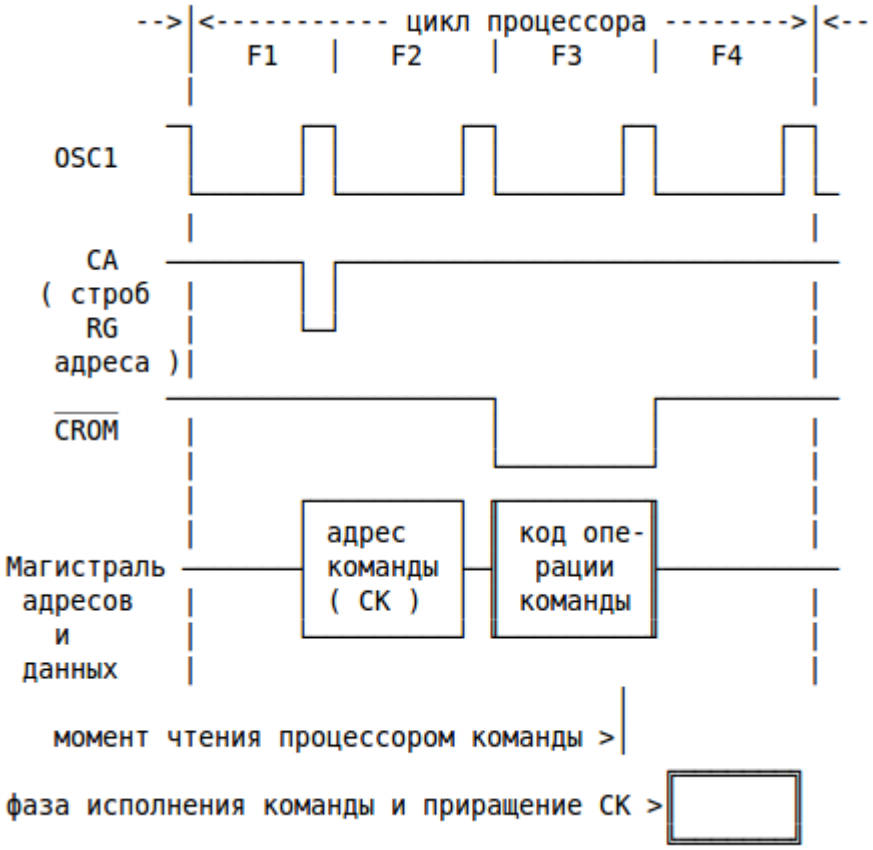
4. Порты, выдающие логическую "1", могут управлять резистивной нагрузкой величиной свыше 2 кОм (вытекающий ток до 2 mA).

5. ТТЛ и NMOП схемы к выходам внутренних портов процессора подключать через буферные КМОП-ТТЛ усилители, например, К561ПУ4, а к входам - непосредственно, но между выходом ТТЛ или NMOП схемы и выводом источника питания +Ucc необходимо, с целью увеличения логического перепада, установить резистор сопротивлением 10 кОм -:- 33 кОм.

6. На незадействованные выводы внутренних портов нужно в начале программы подать "1" в соответствующие разряды или эти выводы через резисторы сопротивлением около 300 кОм подключить к общему выводу источника питания.

Нагрузочные характеристики всех выводов БИС в режиме "выход" приведены на последних страницах этого описания. Максимальное значение тока утечки на входах "R" и "TEST" не превышает 5 мкА при температуре ниже +30 градусов Цельсия и 15 мкА при +70 градусах. Максимальное значение тока утечки на всех входах-выходах в режиме "вход" не превышает 15 мкА при температуре ниже +30 градусов Цельсия и 50 мкА пРИ +70 градусах.

9 Временная диаграмма выполнения одноцикловой команды



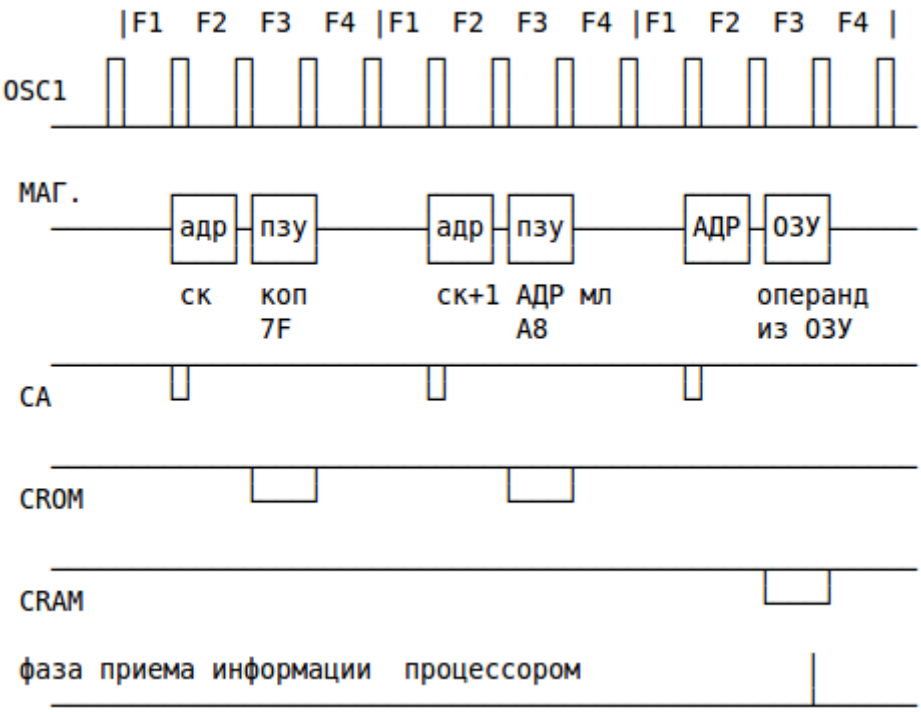
Средним уровнем сигнала на магистрали условно показано "третье" - высокоимпедансное состояние 8-ми младших разрядов адресов-данных.

Многоцикловые команды, кроме команд с ОЗУ, выполняются аналогично, только во втором и третьем циклах процессор получает адрес (сначала старший байт адреса) или данные (команда A1xxx). Момент чтения процессором операнда из ОЗУ или внешнего порта — передний фронт OSC1 в фазе F3.

Задержки всех выходных сигналов (кроме выводов портов P00-:-P33) при Сн не более 50 пФ относительно фронтов сигнала на входе OSC1 не превышают 250 нсек. Измерения производить при условии, что выход OSC2 нагружен емкостью не более 10 пФ.

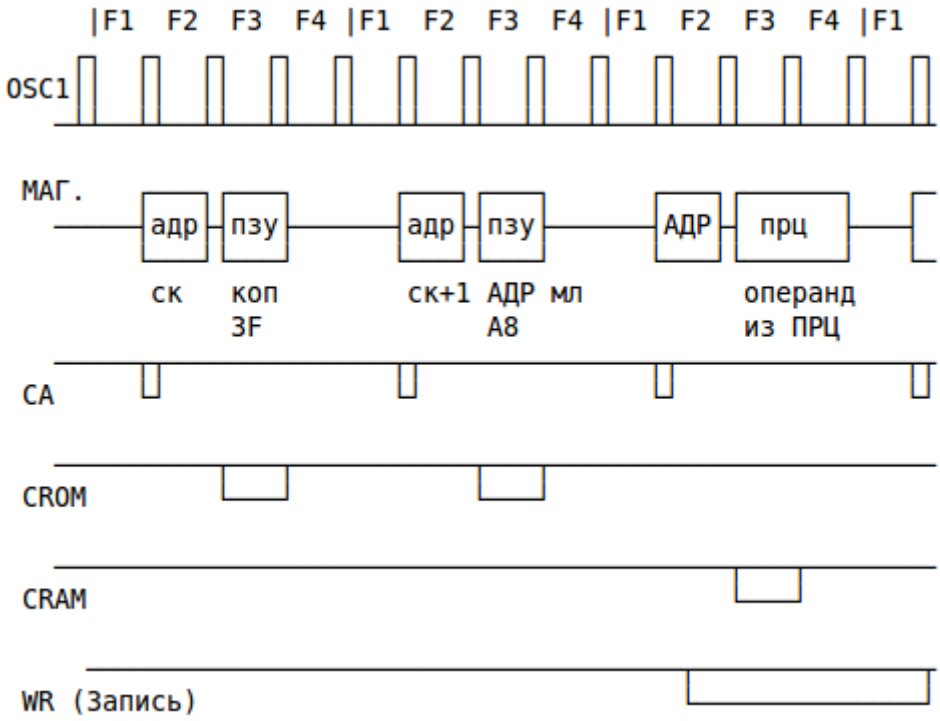
10 Временная диаграмма выполнения команды чтения из ОЗУ

Пример: команда 7F A8 - чтение из ячейки с адресом A8. (WR = 1)



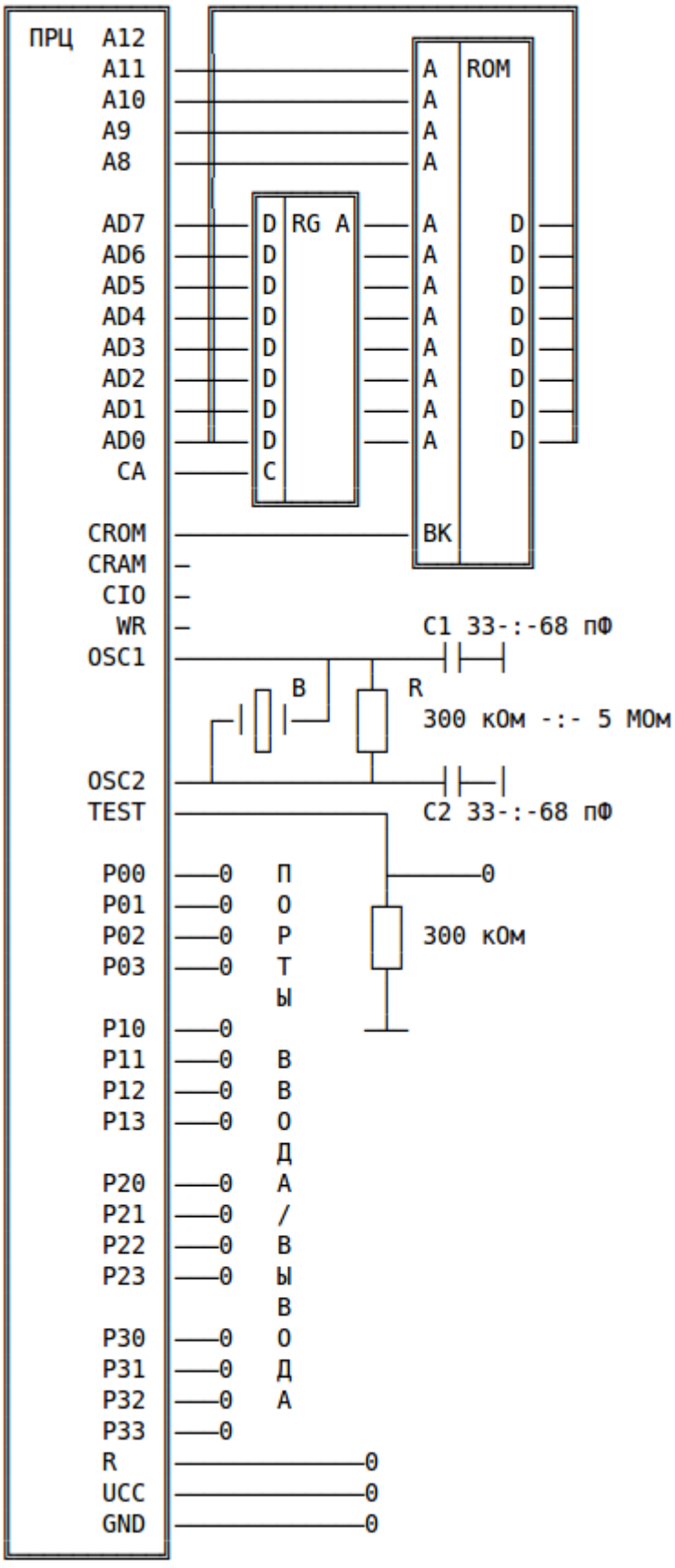
11 Временная диаграмма выполнения команды записи в ОЗУ

Пример: команда 3F A8 - запись в ячейку с адресом A8.



Аналогично происходят обращения к внешним портам ввода/вывода, только вместо выхода "CRAM" активен выход "CIO".

12 Минимальная конфигурация



Регистр адреса (RG A): К564ИР6 или К564ИР9, К564ТМ3 - по 2 шт.

ПЗУ (ROM): NMOП: К573РФ2, К558РР3; КМОП: К563РЕ1, К1603РЕ1, К1623РТ1, К1626РФ1.

В микросхеме К558РР3 можно использовать встроенный адресный регистр, следовательно можно обойтись двумя БИС.

13 Системы отладки программ

То, что в начале фазы F1 магистраль процессора находится в "третьем" состоянии позволяет при подаче на вход "R" микросхемы логической "1" внешним устройствам свободно работать через магистраль с ОЗУ и ПЗУ. Это свойство и используется в отладочных системах.

Предлагается 3 способа отладки программ, различающихся по количеству необходимого оборудования и по удобству работы:

1. Самый простой способ отладки программ на процессоре КР1806ВЕ1: проверенную с помощью кросс-интерпретатора программу записать в БИС УФСППЗУ типа К573РФ2, к магистрали процессора через усилители тока на транзисторах подключить светодиоды. Тактовые импульсы от кнопки "ТАКТ" через схему устранения дребезга и одновибратор с длительностью импульса менее 25 мкс надо подавать на вход "OSC1" процессора и наблюдать на светодиодном индикаторе адреса и команды, появляющиеся на магистрали в процессе выполнения программы. Для ускорения попадания системы в нужное состояние можно сделать генератор пачек тактовых импульсов. Электрическая принципиальная схема такого устройства приведена на иллюстрации, получаемой при запуске на исполнение файла otl_sys.bat в подсправочнике OTL (микросхемы D55 и D56).

2. ЭВМ через выходной порт, например, порт принтера или RS-232, загружает в ОЗУ отладочной системы - эмулятор программного ПЗУ отлаживаемую программу. Трассировка происходит аналогично предыдущему способу. Преимущество этого варианта состоит в существенном уменьшении времени коррекции программы.

3. Удобнее анализировать процесс выполнения отлаживаемых программ, если есть возможность сохранять файлы трассировки в ЭВМ. Для этого ЭВМ должна иметь, как минимум, два 8-ми разрядных порта ввода-вывода:

1) двунаправленный порт, подключенный к магистрали, используется для записи отлаживаемой программы из ЭВМ в ОЗУ и для передачи состояний магистрали в ЭВМ в режиме трассировки;

2) порт управления отладочным комплексом и синхронизации процессов в микропроцессоре и отладочной системе.

Электрическую принципиальную схему этого варианта отладочной системы можно распечатать, запустив на исполнение файл otl_sys.bat в подсправочнике \OTL переданной Вам дискеты.

14 Таблицы параметров БИС КР1806ВЕ1 при приемке и поставке

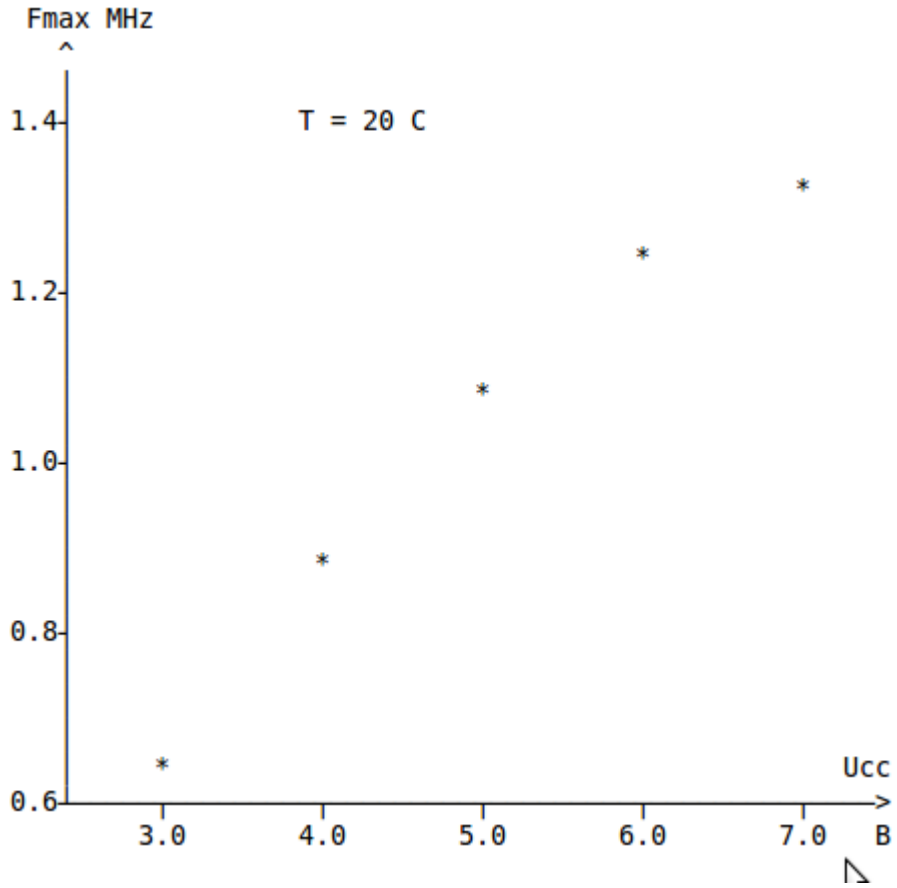
при Ucc = 5+-10% (из ТУ)

НАИМЕНОВАНИЕ ПАРАМЕТРА, ЕДИНИЦА ИЗМЕРЕНИЯ	НОРМА		ТЕМПЕРА-ТУРА, С
	НЕ МЕНЕЕ	НЕ БОЛЕЕ	
ВЫХОДНОЕ НАПРЯ-ЖЕНИЕ НИЗКОГО		0,5	25+-10
УРОВНЯ,В		0,7	-25, 70
ПРИ Iol = 0,5 мА			
ВЫХОДНОЕ НАПРЯ-ЖЕНИЕ ВЫСОКОГО	3,6		25+-10
УРОВНЯ,В	3,4		-25, 70
ПРИ Iон = 0,5мА			
ТОК ПОТРЕБЛЕНИЯ, мА		0,25	25+-10
ПРИ Fт = 30кГц		1,0	-25, 70
ТОК УТЕЧКИ НА ВХОДЕ,мкА		5	25+-10
		15	-25, 70
ТОК УТЕЧКИ НА ВХОДЕ-ВЫХОДЕ,		15	25+-10
мкА		50	-25, 70
МАКСИМАЛЬНАЯ ТАКТОВАЯ	1,0		25+-10
ЧАСТОТА,мГц	0,8		-25, 70
ПРИ Сн=50пФ			

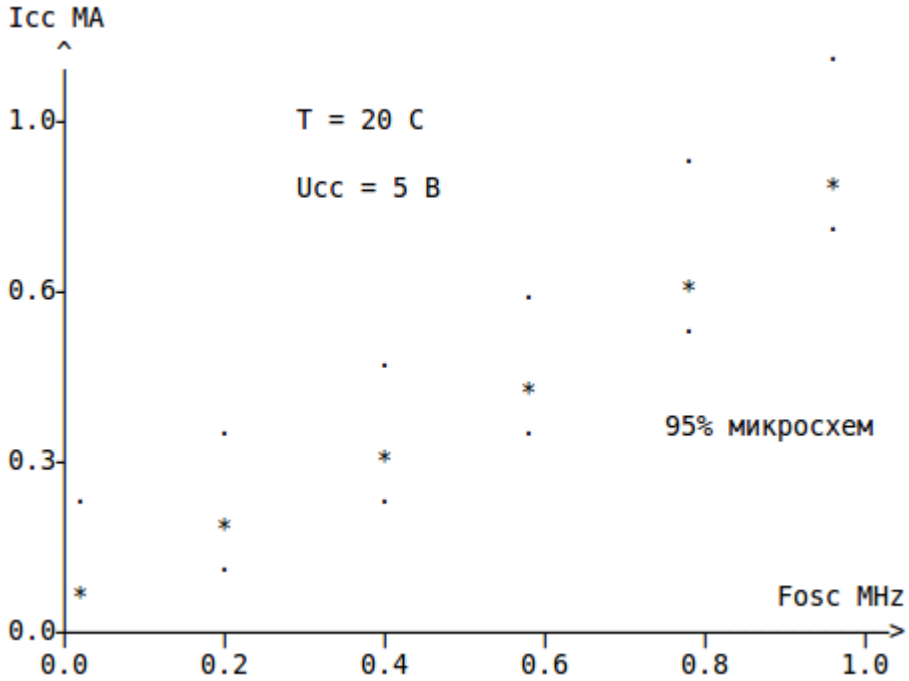
15 Предельно-допустимые параметры БИС КР1806ВЕ1

НАИМЕНОВАНИЕ ПАРАМЕТРА,ЕДИНИЦА ИЗМЕРЕНИЯ	НОРМА	
	НЕ МЕНЕЕ	НЕ БОЛЕЕ
НАПРЯЖЕНИЕ ПИТАНИЯ,В	4,0	6,5
ВХОДНОЕ НАПРЯЖЕНИЕ ВЫСОКОГО УРОВНЯ,В	0,75 Ucc	Ucc+0,3
ВХОДНОЕ НАПРЯЖЕНИЕ НИЗКОГО УРОВНЯ,В	-0,3	0,25 Ucc
ВЫХОДНОЙ ТОК, мА		2
ЕМКОСТЬ НАГРУЗКИ,пФ		200

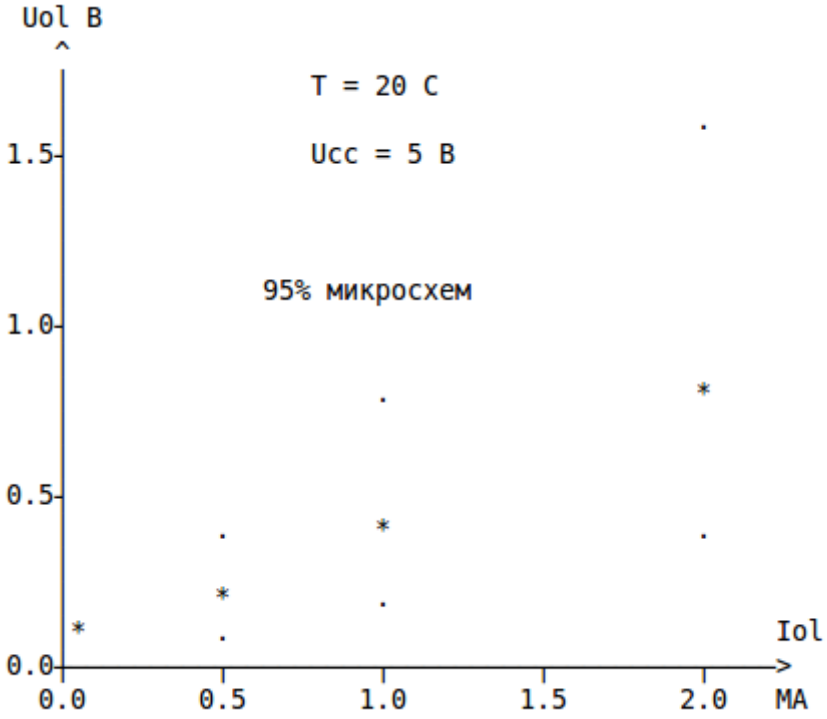
16 Зависимость максимальной тактовой частоты функционирования процессора (Fmax) от напряжения питания (Ucc)



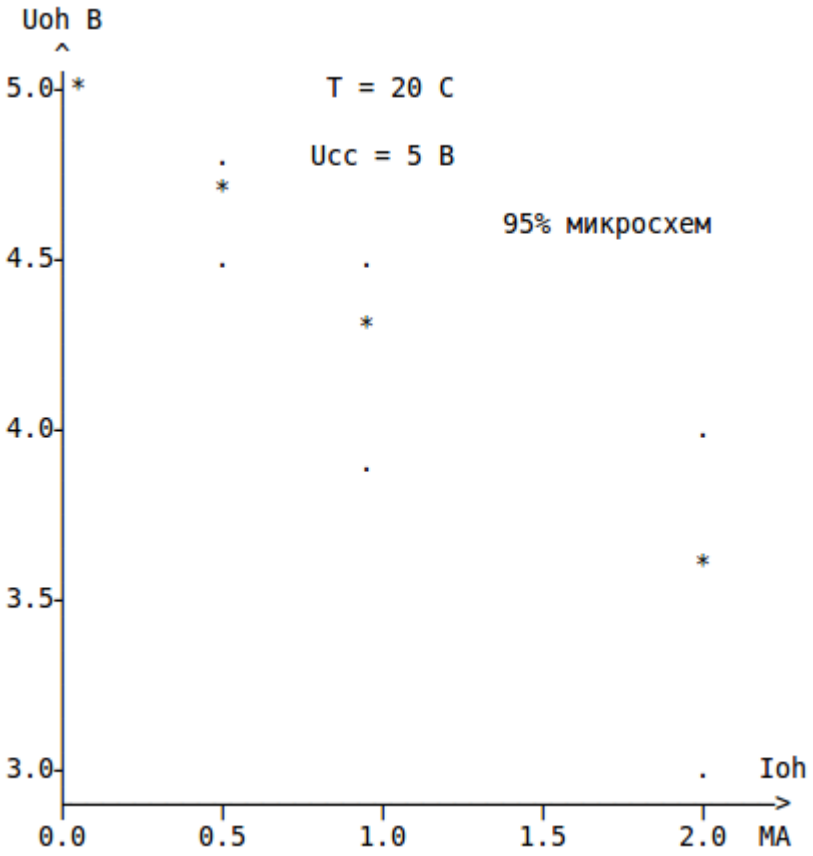
17 Типичная зависимость тока потребления (Icc) от тактовой частоты (Fosc)



18 Зависимость выходного напряжения низкого уровня (Uol) от выходного тока (Iol) для выходов, входов-выходов магистрали



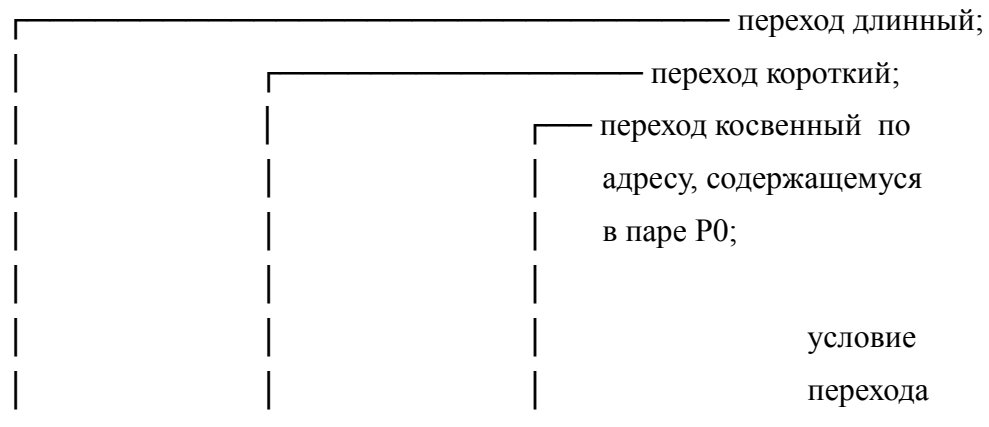
19 Зависимость выходного напряжения высокого уровня (Uoh) от выходного тока (Ioh) для выходов, входов-выходов магистрали и внутренних портов ввода-вывода



Приложение 1. Кодировка команд ассемблера

Количество "решеток"- # после мнемоники команд указывает число байтовых переменных требуемых ассемблером, а в скобках приведены коды операций в шестнадцатиричной системе счисления.

Команды переходов



jmp##[00/08]	brn#[10/18]	kbr[20/28]	безусловный
jc0##[01/09]	bc0#[11/19]	kc0[21/29]	при "C"=0
jc1##[02/0A]	bc1#[12/1A]	kc1[22/2A]	при "C"=1
jz0##[03/0B]	bz0#[13/1B]	kz0[23/2B]	при "Z"=0
jz1##[04/0C]	bz1#[14/1C]	kz1[24/2C]	при "Z"=1
jtg##[05/0D]	btg#[15/1D]	ktg[25/2D]	при "ТГТ"=1
jvt##[06/0E]	bvt#[16/1E]	kvt[26/2E]	при "TEST"=1

После наклонной черты - коды команд перехода на подпрограмму.

Команды переходов на подпрограмму кодируются аналогично обычным командам перехода с добавлением литеры "s".

Пример:

sbz1# - команда перехода на подпрограмму при "Z"=1.

Безадресные команды

ret [34] возврат из подпрограммы
rap [38] RG адреса прерывания -> СК
nop [44] нет операции (НОП)
hlt [45] останов процессора
aps [46] режим аппаратного стека
prs [47] режим программного стека
sti [54] запуск таймера
rti [55] останов таймера
rtc [56] "0" -> "С"; установка в "0" триггера "С"
stc [57] "1" -> "С"; установка в "1" триггера "С"
rja [64] режим "АР",сброс"+С" И "ЛОГ"
rjc [65] режим "+С"
rjl [66] режим "ЛОГ"
sdv [74] цикл. сдвиг аккумулятора вправо
tg0 [75] "0" -> "ТГТ"; обнуление триггера таймера
ivc [76] (не"С") -> "С"; инверсия триггера "С"
iva [77] (не А) -> А ; инверсия аккумулятора

Команды обращения к ОЗУ и регистрам

mXX [3X]   чтение аккумулятора
cXX [4X]   сравнение с аккумулятором (caXX в лог. режиме)
sXX [5X]   вычитание из аккумулятора (anXX в лог. режиме)
aXX [6X]   сложение с аккумулятором (m2XX в лог. режиме)
lXX [7X]   запись в аккумулятор
/ XX /   способ адресации

Xr0 [X0]   регистр R0
Xr1 [X1]   регистр R1

Xr2 [X2] регистр R2

Xr3 [X3] регистр R3

X00 [X9] ОЗУ косв. адресация по P0, "0" страница

X01 [XA] ОЗУ косв. адресация по P1, "0" страница

Xk0 [XB] ОЗУ косв. адресация по P0

Xk1 [XC] ОЗУ косв. адресация по P1

X0s# [XD] ОЗУ "0" страница

X0p# [XD] порты ("0" страница, адрес - 0..F/H)

Xdl##[XE] ОЗУ длинная адресация

Xkr# [XF] ОЗУ короткая адресация

Пример:

s0s# - команда вычитания из аккумулятора слова, находящегося в "0" странице ОЗУ.

Команды обращения к парам регистров

p0X [8X] P0 -> PX (X = 0 -:- B) запись из пары P0 в любую пару регистров;

pX0 [9X] PX -> P0 (X = 0 -:- B) запись из любой пары регистров в пару P0;

p1X [AX] P1 -> PX (X = 0 -:- 7) запись из пары P1 в любую пару регистров;

pX1 [BX] PX -> P1 (X = 0 -:- 7) запись из любой пары регистров в пару P1;

psX# [AY] D8 -> PX (X = 0 -:- 7; Y = X + 8) запись второго байта команды в любую пару регистров;

roX [BY] ПЗУ(P0) -> PX (X = 0 -:- 7; Y = X + 8) запись байта из ячейки ПЗУ с адресом - содержимым пары P0 в любую пару регистров;

D8-константа(2-й байт команды).

Команды p00,p11,p40,p60,p41 и p61 запрещены.

Команды с непосредственной адресацией

cdX [CX] сравнение константы с аккумулятором(cadX в лог.режиме)

sdX [DX] вычитание константы из аккумуляторА(andX в лог.режиме)

adX [EX] сложение константы с аккумулятором (m2dX в лог.режиме)

dka [E6] десятичная коррекция аккумулятора (m2d6 в лог.режиме)

ldX [FX] запись константы в аккумулятор

Приложение 2. Программный комплекс, предназначенный для разработки пользовательских программ, исполняемых на микропроцессоре КР1806ВЕ1

Программы предназначены для работы на микро-ЭВМ, совместимых с IBM PC/XT, AT с версией MSDOS не хуже 3.0. Возможна поставка части комплекса (без Turbo-среды, редактора и встроенной программы "Help") для микро-ЭВМ типа ДВК.

Программа кросс-транслятор с языка ассемблера микропроцессора КР1806ВЕ1

Программа преобразует пользовательскую программу, написанную на языке ассемблера процессора КР1806ВЕ1, в "исполняемый" модуль.

Имя входного файла : '*.ass'. (* - любое разрешенное MSDOS имя)

Выходные файлы -"исполняемый" модуль '*.dat' и бинарный файл '*.cod'для зашивки в ПЗУ.

Описание операторов языка ассемблера:

; - признак комментария до конца строки;

;; - (в начале строки) - признак наименования модулей программы для использования в конденсированной форме вывода файла на экран (количество модулей в программе - не более 50);

{ } - скобки для комментария-блока из нескольких строк (символы '{' и '}' должны быть единственным символами в строке и находиться в первом столбце строки).

***** - в начале строки - признак определения адреса переменной или метки или команд ;

= - определение значения переменной, адреса переменной или метки;

: - использование метки;

[ ] - использование аналога массива (см. далее);

segment - в начале строки - признак перехода на следующую страницу ПЗУ;

***fill=0FF** - специальная переменная для заполнения промежутков между модулями с физической адресацией (в данном случае ПЗУ заполняется кодом FF/hex). По умолчанию ПЗУ заполняется командой nop - 44/hex

***sub_test = 1** - в этом случае компилятор проверяет наличие символа "_" (подчеркивание) в именах подпрограмм и отсутствие его в метках обычных переходов - эта команда позволяет уменьшить количество ошибок в программах пользователей, при = 0 - контроль отключен (по умолчанию);

***warn_def = 1** - команда подавляет выдачу предупреждений "переменная использована раньше, чем определена";

Названия команд запрещены для наименования пользовательских переменных и меток.

Определение адресов переменных, значений констант и т.д. производится в шестнадцатиричном коде, но, для различения шестнадцатиричных чисел и имен переменных, начинающихся с букв "a","b","c","d" и "f", необходимо перед значением числа поставить цифру "0", например: ***ba = 0A7** - здесь переменной ba присвоено значение A7.

В программах на языке ассемблера можно использовать конструкцию - аналог массива: например, Вам нужно сложить 16 ячеек ОЗУ с адресами от 0160 до 016F. В ранних версиях компилятора Вам пришлось бы определить имена всех 16-ти переменных: pr0-:-prF и "потратить" на эти определения 16 строк ассемблерного файла. Теперь Вам достаточно определить только адрес начала массива:

*** pr = 0160,**

а использовать эти 16 переменных можно таким образом:

adl pr[0] (соответствует адресу 0160);

ps1 pr [1] (соответствует адресу 0161);

. . .

a0s pr[0E] (соответствует адресу 016E);

mkr pr[0F] (соответствует адресу 016F);

При этом адреса в файле *.dat будут получены в результате сложения начального адреса и параметров массива.

Пример 1: входной файл **a.ass**:

```
* Ku_ku = 2a  ; "=" - определение Ku_ku ='2a' (байт)
* Label = 0110 ; "=" - определение Label = '0110'(2 байта)
;; модуль 1    ; начало модуля 1
0000:    nop   ; установка адреса 0000;
t:       jmp pl    ; ":" - метка   't';
bz0 t
segment        ; переход на следующую страницу (здесь 01);
* fill = 45  ; - определение байта заполнения ПЗУ
nop
pl:      ldF       ; ":" - метка  'pl';
sd1
;; таблица        ; модуль "таблица"
Label:           ; установка адреса на 0110;
; запись таблицы в ПЗУ:
Ku_ku       ;    байт '2a' по адресу 0110;
0f1         ;    байт 'f1' по адресу 0111;
```

Пример 2: Входной файл **b.ass**:

```
* uu = 0d0
bz0  pl
```

```
mdl 456a
akr uu[5]
segment
hlt
sjmp  pl
```

Порядок перечисления в файле программных модулей обязательно должен соответствовать порядку размещения программных модулей в ПЗУ, т.е. нельзя ставить перед модулем с меньшим физическим адресом модуль с большим адресом. После слияния 2-х файлов и запуска : mp4_comp ab.ass получаем выходной файл ab.dat (здесь ... - упущены коды заполнения ПЗУ):

```
44-0000          a.ass
00-0001          ---------------
01-0002          |* Ku_ku = 2a
01-0003          |* Label == 0110
13-0004          |;; модуль 1  ; начало модуля 1
01-0005          |0000:    nop
44#0006          |t:       jmp pl
44#0007          |         bz0 t
…                |segment
44#00FE          |* fill = 45
44#00FF          |         nop
44-0100          |pl:      ldF
FF-0101          |         sd1
D1-0102          |;; таблица  ; модуль "таблица"
45#0103          |Label:
45#0104          |         Ku_ku
 ...             |         0f1
45#010E          ---------------
45#010F
2A-0110
F1-0111          b.ass
13-0112          -------------------
01-0113          |* uu = 0d0
3E-0114          |         bz0  pl
45-0115          |         mdl 456a
6A-0116          |         akr uu[5]
6F-0117          |segment
D5-0118          |         hlt
45#0119          |         sjmp  pl
45#011A          -------------------
...
45#01FE
45#01FF
45-0200
08-0201
01-0202
01-0203
```

Признак заполнения ПЗУ - '#' в третьем столбце выходного файла. По умолчанию заполнение "промежутков" в адресах между модулями программы производится кодом 44 (NOP). Правее загрузочного модуля для сравнения приведен исходный текст программы.

Программа кросс-интерпретатор

Программа " mp4_int.exe " предназначена для моделирования процессора КР1806ВЕ1 и получения инфермации о процессе исполнения пользовательской программы, записанной в шестнадцатиричных машинных кодах в исходный для этой программы файл YYY.dat. (YYY - произвольное имя файла).

Процесс исполнения пользовательской программы выдается на экран ЭВМ и одновременно записывается в файл результата - YYY.rez.

Пример исходного файла:

Файл dd.dat:

```
 ---------------------------------------------
|64-0000     | установка режима "АР".
|46-0001     | Аппаратный стек.
|F3-0002     | запись константы 3 в А.
|;<- признак комментария.
|3e-0003     | запись А в ОЗУ по адресу 0164
|01-0004     |   (длинная адресация ).
|64-0005     |
|e2-0006     | сложение А с константой 2.
|1E-0007     | короткий переход на подпрог-
|0c-0008     | рамму на 0С при "TEST" = 1.
|00-0009     | безусловный длинный пере-
|00-000A     |  ход на 0006.
|06-000B     |
|A8-000C     | запись байта 64
|64-000D     |  в пару регистров Р0.
|5b-000E     | вычитание из А содержимого
|;           |  ячейки ОЗУ с адресом,
|;           |  содержащемся в паре Р0.
|34-000F     | возврат из подпрограммы.
|---------------------------------------------
 1  колонка файла
  2  колонка файла
   3  колонка файла
```

Программа интерпретатора анализирует только первые 3 колонки файла:

- 1 и 2 колонки - коды команд;

- 3-я колонка обязательно должна быть в файле, но значение ее произвольно, но символ " R " обнуляет программный счетчик прецессора, символ " T " - моделирует подачу на вход

"TEST" сигнала "1", символ " N " - "0".

В следующих колонках файла могут быть записаны адреса команд в ПЗУ, номер соответствующей строки исходного файла и комментарии.

Символ " ; " в 1-й колонке показывает, что вся строка - комментарий.

Исходный файл может быть набран в любом редакторе, но первые 3 колонки обязательно латинским алфавитом прописными или строчными буквами.

В комментариях можно использовать кириллицу.

Запуск программы mp4_int может производиться только(!) программой mp4_.exe из интегрированной среды.

Директивы интерпретатора:

H - Help - вызов такого-же списка команд интерпретатора;

Q - вызывает Help-а по процессору;

1, **4**, **8** и **F** - выводит на экран результат работы процессора за 1, 4, 8 и 16 циклов соответственно;

A - выводит на экран результат работы процессора до тех пор, пока счетчик циклов или счетчик команд не будут кратны 100;

S - выбор номеров 2 и 3-й страниц ОЗУ для вывода на экран и записи в файл результатов.

O - выводит на экран 3 страницы ОЗУ;

M - записывает в файл результата текущее состояние 3-х страниц ОЗУ;

T/N - директива"T"- устанавливает на вход процессора "TEST" сигнал "1", а директива "N"- "0"(по умолчанию);

+/- - директива"+"- устанавливает режим формирования файла результатов (по умолчанию), а "-" - отключает его;

P - записывает в порт шестнадцатиричную константу;

C - выдает на экран результат работы процессора до указанного пользователем номера машинного цикла процессора;

Z - прогон программы до тех пор, пока не будут выполняться команды "aps" или "prs"(обычно эти команды стоят только в начале программы) - аналог точек останова;

L - прогон программы до выбранной строки (Enter — повтор);

(Команды C,Z и L можно остановить, нажав клавишу "пробел")

K или **Esc** - директива окончания работы программы.

Далее приведена распечатка файла результата, причем на следующей строке после каждой команды знаками " ^ " и " #### " отмечены изменения в регистрах процессора.

Если в какой-либо регистр пока не происходила запись, то состояние его обозначается литерой " X "- произвольное значение. В конце работы программы в файл результата

производится запись содержимого 3-х страниц ОЗУ (знаком "." отмечены ячейки ОЗУ в которые не производилась запись).

Файл dd.rez:

```
CTC                R  A       AM RRRR                        U          RT R
     AD  DA IN JT K CZ P4   0123 P2 P3 P8P9 STE2 S P6 P7TG S PAPB
   1-0000-64-64-A0-*-X0-XX-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-X-XXXX
                     ^
   2-0001-46-46-A0-*-X0-XX-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
                                                                  ^
   3-0002-F3-F3-A0-3-X0-XX-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
                       ^
   4-0003-3E-3E-A0-3-X0-XX-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
   5-0004-01-3E-A0-3-X0-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
   6-0005-64-3E-A0-3-X0-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
   7-0164**3 3E-A0-3-X0-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
      ^^^^   ^                  ^^
   8-0006-E2-E2-A0-5-00-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
                       ^
   9-0007-1E-1E-A0-5-00-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
  10-0008-0C-1E-A0-5-00-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
                                      # перехода нет
  11-0009-00-00-A0-5-00-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
  12-000A-00-00-A0-5-00-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
  13-000B-06-00-A0-5-00-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
      ####         переход !
  14-0006-E2-E2-A0-7-00-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
                       ^
  15-0007-1E-1E-A0-7-00-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
  16-0008-0C-1E-A0-7-00-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
            #перехода нет
  17-0009-00-00-A1-7-00-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
  18-000A-00-00-A1-7-00-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
  19-000B-06-00-A1-7-00-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
                     ^   !!!!
  20-0006-E2-E2-A1-9-00-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
                       ^
  21-0007-1E-1E-A1-9-00-01-XXXX-XX-XX-XXXX-XXXX-0-00-XX0X-A-XXXX
  22-0008-0C-1E-A1-9-00-01-XXXX-XX-XX-0009-XXXX-1-00-XX0X-A-XXXX
      ####           ^                          ^^^^          ^
                                 переход на п/п при "TEST" = 1
  23-000C-A8-A8-A1-9-00-01-XXXX-XX-XX-0009-XXXX-1-00-XX0X-A-XXXX
  24-000D-64-A8-A1-9-00-01-64XX-XX-XX-0009-XXXX-1-00-XX0X-A-XXXX
                                 ^^
  25-000E-5B-5B-A0-9-00-01-64XX-XX-XX-0009-XXXX-1-00-XX0X-A-XXXX
  26-0164**3 5B-A0-6-10-01-64XX-XX-XX-0009-XXXX-1-00-XX0X-A-XXXX
         ^^  ^
  27-000F-34-34-A0-6-10-01-64XX-XX-XX-0009-XXXX-0-00-XX0X-A-XXXX
      ####         возврат из п/п                              ^
  28-0009-00-00-A0-6-10-01-64XX-XX-XX-0009-XXXX-0-00-XX0X-A-XXXX
```

```
 29-000A-00-00-A0-6-10-01-64XX-XX-XX-0009-XXXX-0-00-XX0X-A-XXXX
 30-000B-06-00-A0-6-10-01-64XX-XX-XX-0009-XXXX-0-00-XX0X-A-XXXX
     ####        длинный переход
 31-0006-E2-E2-A0-8-00-01-64XX-XX-XX-0009-XXXX-0-00-XX0X-A-XXXX
                      ^
  %%% STR"0" %%%  %%% STR"01" %%%  %%% STR"1E" %%%
 0* 0000.... ........ * ........ ........ * ........ ........ *
 1* ........ ........ * ........ ........ * ........ ........ *
 2* ........ ........ * ........ ........ * ........ ........ *
 3* ........ ........ * ........ ........ * ........ ........ *
 4* ........ ........ * ........ ........ * ........ ........ *
 5* ........ ........ * ........ ........ * ........ ........ *
 6* ........ ........ * ....3... ........ * ........ ........ *
 7* ........ ........ * ........ ........ * ........ ........ *
 8* ........ ........ * ........ ........ * ........ ........ *
 9* ........ ........ * ........ ........ * ........ ........ *
 A* ........ ........ * ........ ........ * ........ ........ *
 B* ........ ........ * ........ ........ * ........ ........ *
 C* ........ ........ * ........ ........ * ........ ........ *
 D* ........ ........ * ........ ........ * ........ ........ *
 E* ........ ........ * ........ ........ * ........ ........ *
 F* ........ ........ * ........ ........ * ........ ........ *
  * 01234567 89ABCDEF * 01234567 89ABCDEF * 01234567 89ABCDEF *
```

Адрес ячейки ОЗУ в странице определяется следующим образом: старшая часть адреса в странице - символ по вертикали, а младшая - по горизонтали (на примере: в 1-й странице ОЗУ число "3" записано по адресу 0164/H).

Заголовок файла **dd.rez**:

```
CTC            R  A      AM  RRRR                                U          RT R P P
  AD DA IN JT K CZ  P4   0123 P2 P3 P8P9 STE2 STE3  S P6  P7TG S A B
```

CTC - счетчик циклов, смоделированных программой;

AD - адрес выдаваемый процессором;

DA - информация на шине данных : команды, константы из ПЗУ и операнды из ОЗУ и для ОЗУ;

IN - выполняемая команда (инструкция);

RJ - режим процессора "АР" - A, "+C" - C и "ЛОГ" - L;

T - состояние входа "TEST" - 0 или 1;

AK - аккумулятор;

C - триггер-флажок "C";

Z - флажок "Z";

AM P4 - адрес рабочей страницы ОЗУ P4;

R0 R1 R2 R3 - 4-х разрядные регистры R0 -:- R3, они же составляют пары регистров

P0 и P1;

P2 P3 - 8-ми разрядные пары регистров P2 и P3;

P8 P9 - стек 1;

STE2 STE3 - стеки 2 и 3;

US - указатель уровня рабочего стека - 0(исходное состояние) или 1, 2, 3;

P6 - регистр управления таймером P6;

P7 - таймер-счетчик P7;

RT - режим таймера: 0 - таймер выключен, 1 - включен;

TG - триггер таймера - "TГТ";

RS - указатель режима стека A - аппаратный, P - программный;

PA PB - регистр адреса прерывания - пары PA и PB;

В последних версиях программы добавлена колонка номера строки исходного файла ассемблера.

При чтении ячеек ОЗУ, в которые еще не производилась запись, программа выводит сообщение об ошибке перед неправильно использованной командой.

При отладке программ "реального времени", например, прием информации по каналу RS232, возникают большие трудности по моделированию сложных входных сигналов. Для облегчения отладки таких программ предусмотрен режим ввода сигнала, поступающего на вход "TEST": в текстовый файл, совпадающий по имени с *.ass, - *.tes запишите символы "1" и "0", причем каждой единице соответствует 10 циклов "1" на входе "TEST" и нулю - 10 циклов "0". По строкам файл можно форматировать произвольно. Если в рабочем справочнике есть файл *.tes, совпадающий по имени с рабочим *.ass, то входные воздействия на вход "TEST" берутся только (!) из файла *.tes . Если входные воздействия должны поступать на вход порта - сделайте вариант программы с приемом сигнала на вход "TEST" (добавьте нужное количество NOP-ов для выравнивания времени) и уже проверенную программу переделайте на нужный вход.

Интерпретатор существенно облегчает отладку программ пользователей, т.к. при работе программы легко контролируются режимы, в которых находится процессор, а также регистры и ячейки внутреннего ОЗУ БИС. Совпадение работы программы и функционирования процессора максимально возможное, потому что разработчик БИС и программной модели в данном случае одно и то же лицо. Результат отладки программы с помощью интерпретатора: даже в первый раз в ППЗУ записывается в большой степени функционирующая программа, а ошибки обычно видны "невооруженным" взглядом.

Комплекс программ отладки

Программа mp4_.exe: Turbo-подобная среда по отладке пользовательских программ для процессора КР1806ВЕ1. Она объединяет программы ассемблера и интерпретатора. Для работы, например, с программой p1.ass нужно набрать команду: **C:>mp4_ p1.ass**

Таким образом Вы попадаете во встроенный редактор, который может исполнять следующие команды:

Ctrl+Y - стирание строки;

Ctrl+G - переход на строку с введенным номером;

Ctrl+B - маркировка начала блока;

Ctrl+E - маркировка конца блока, отмена блока;

Ctrl+C - копирование блока;

Ctrl+D - удаление блока;

Ctrl+V - перенос блока;

Ctrl+"->" - переход на слово влево;

Ctrl+"<-" - переход на слово вправо;

Ctrl+"-" - установка маркера;

Alt +"-" - возврат на маркер;

Ctrl+L - символ конца страницы;

Ctrl+F - поиск строки;

Ctrl+A - продолжение поиска строки;

Ctrl+Z - конденсированная форма файла;

!!!!! Используйте этот режим при разработке больших и средних программ !!!!!

F1 - помощь по встроенному редактору и ассемблеру;

Shift+F1 - помощь по процессору ;

Ctrl +F1 - контекстная помощь по командам процессора;

F2 - запись файла;

Ctrl+F2 - запись файла *.lin с номерами строк для печати файла при отладке;

F3 - загрузка файла;

F10 - вызов меню;

Alt+X - выход из редактора.

Команды системы:

Вызов программы компилятора:	**Alt + F9**;
Вызов программы интерпретатора:	**F9**;
Вызов программ компилятора и (если нет ошибок) интерпретатора:	**Ctrl + F9**;
Просмотр листинга ошибок:	**F6;**
Просмотр экрана результата:	**Alt + F5**;

Размер файла *.ass в редакторе mp4_.exe ограничен 60 Кб., но, если Вам надо обработать файл превышающий эту величину, запустите программу mp4_k.exe, которая удалит из файла ассемблируемой программы комментарии и запишет новый файл (к старому имени файла добавится "1").

Файл *.lin удобно распечатать и работать именно с ним при отладке программ пользователей т.к. он показывает номера строк исходного файла *.ass, а они содержатся в файлах *.dat и *.rez.

Большие файлы результатов удобно просматривать не выходя из системы, если использовать из главного меню "Tools" (можно вызывать и Alt+T) систему "Viewer"- там всегда присутствует заголовок файла результата.

Работа системы в режиме проекта.

Возможно собрать программу из нескольких файлов для чего надо в файле **.pr перечислить нужные файлы в правильном порядке (в каждой строке - одно название файла).

Сборка проекта согласно файлу **.pr происходит по команде: - **Shift + F9**.

Пример файла **arifm.pr**:

```
| add1.ass
| sub1.ass
| multipl.ass
| div.ass
| monitor.ass
|
```

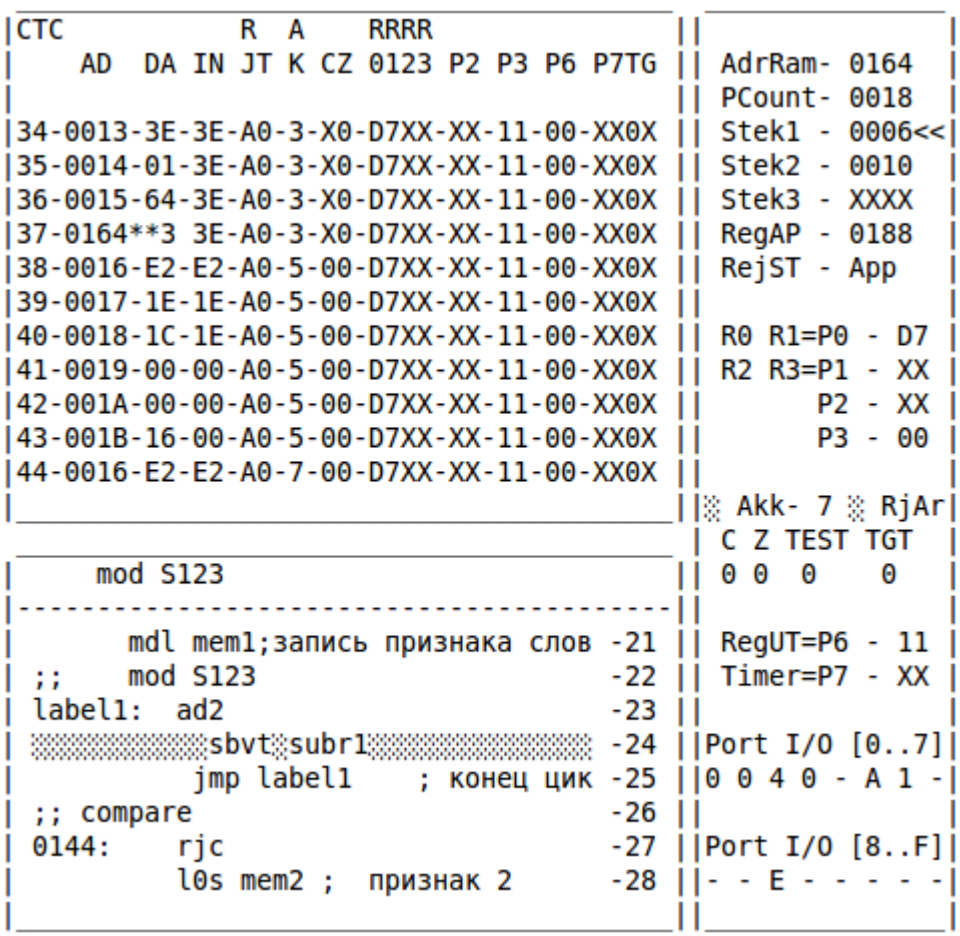
Имя проекта не должно совпадать с именем какого-либо файла *.ass !

С полученным таким образом файлом **.ass можно делать все кроме редактирования. Если же появилась необходимость редактировать, то нужно в месте изменения нажать "Enter"- Вы окажетесь в нужном Вам исходном файле.

В системе программ есть возможность условного компилирования: в исходном файле с помощью скобок, вызываемых:

Alt +"1" - открывающая и **Alt +"2"** - закрывающая, можно выделить блок, который будет компилироваться при работе без файла **.pr и не будет при наличии в рабочем справочнике файла **.pr. Это удобно для вызова отлаживаемой программы из разных мест при работе с проектом и без него. При работе в режиме проекта пользуйтесь разделом системного меню "Projekt".

На этой странице показан вид экрана при работе интерпретатора:



На экране можно наблюдать 3 окна:

- верхнее окно показывает процесс работы процессора и практически полностью повторяет результат работы программы mp4_int.exe;

- нижнее окно выделяет строку исходного файла, которая будет интерпретироваться в следующем цикле и показывает имя исполняемого в данный момент модуля;

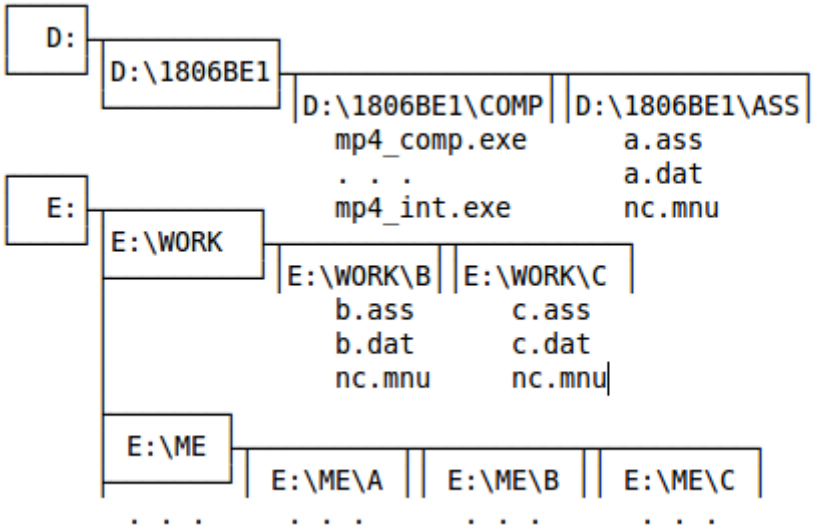
- правое окно, показывающее текущее состояние всех регистров процессора.

В данном примере интерпретируется команда перехода на подпрограмму (строка 24 исходного файла), записанная по адресам 0017 и 0018 и находящаяся в модуле" mod S123 ".

В данный момент программа работает в подпрограмме первого уровня: знак "<<" показывает на Stek1. Если бы предыдущая команда была бы командой сравнения, то флажки "C" и "Z" показывались бы в виде: "#0#1#".

Рекомендуется размещать файлы системы в отдельный подкаталог, а для рабочих файлов пользователей завести собственные подкаталоги для каждого пользователя и даже для отдельных подзадач.

Пример дерева каталогов:



Файл nc.mnu одинаков во всех подкаталогах и в данном примере выглядит следующим образом:

I: Assembler/Interpretator

D:\1806BE1\comp\mp4_

Пользователю в случае использования другого логического диска нужно только изменить путь файла mp4_.exe.

Вход в систему происходит в такой последовательности:

---

1. F2 - вызов UserMenu;

--

Enter -> загружается система отладки

---

2. F3 - Load

--

Enter -> загрузка рабочего файла в систему;

---

3. Поиск нужного файла с помощью клавиш управления положением курсора;

Enter -> Ваш файл загружен в систему.

Работа над комплексом программ продолжается, но уже разработаны и отлажены программы объемом до 70 кбайт исходного текста на языке ассемблера (с комментариями) и до 2 кбайт кода ПЗУ. В систему постоянно включаются дополнительные вспомогательные модули, поэтому, прежде чем использовать новую версию программы, просмотрите и опробуйте все возможные режимы и функции из меню.

Приложение 3. Пример использования микропроцессора КР1806ВЕ1

В качестве примера рассмотрим задачу управления тренировкой аккумуляторов типа Д-0,1. Алгоритм работы этого устройства:

1. аккумулятор в режиме "разряд" разряжаем до 1 В, после чего программа переходит в режим "заряд";

2. аккумулятор в режиме "заряд" заряжаем в течение 15 час.

3. если не включен тумблер "END", идти на п.1, иначе - конец работы устройства.

Электрическая принципиальная схема устройства приведена на странице П4, где обозначены:

G1 - аккумулятор;

VT1 и **VT2** - ключи включения тока заряда и тока разряда, которые определяются резисторами **R1** и **R2**;

VT3,**R5** и **R6** - источник опорного напряжения 1 В;

A1 - компаратор напряжения 1 В и ЭДС аккумулятора;

S1 - тумблер"END", замыкаемый для прекращения процесса тренировки (цикл заряда в этом случае должен завершиться нормальным образом);

VT4..VT10 и светодиодный индикатор **H1** - устройство индикации времени разряда в предыдущем цикле, которое полностью определяет емкость тренируемого аккумулятора.

S2 - кнопка включения индикатора;

D2 и **D1** - тактовый генератор процессора и БИС КР1806ВЕ1;

D3 и **D4** - регистр адреса;

D5 - ПЗУ, в котором записана приведенная ниже программа;

***CW,*RW** и ***VDW** схема "Watchdog" - обнуляет БИС при отсутствии импульсов с порта PI30.

***D6** - схема импульсного питания ПЗУ, снижает ток потребления ПЗУ до величины порядка 1 мА. На рисунке, полученном при запуске ris.bat, приведены временные диаграммы Ugen, Uosc и Ucerom.

* - не обязательные элементы.

```
;; ПРОГРАММА КОНТРОЛЛЕРА ЗАРЯДНО-ТРЕНИРОВОЧНОГО УСТРОЙСТВА.
;#############################################################
* init =    0000;
* ut =  0A7; RG управления таймером: Kd = 30;
;    разрешено прерывание от TGT; вход ПДП подключен к F/4;
* ti =       0FF; таймер
* nul  =      00;
* tabl =    0040;
* raps =      00;  # адрес программы обработки прерываний;
* rapm =      50;  # interr = raps rapm ;
* interr =  0050;  #
* ind1 = 00           ; порты индикатора;
* ind2 = 01           ;
* pupr = 02        ; порт управления;
* wdog = 03 ; порт Watchdog см. описание в файле opus.rus
;;##### инициализация ######################################
init:      rjc        ;
           aps        ;         порт 2: 0-й разряд -
           ps7 ti     ;                 - " 0 " - включен
           tg0        ;                   ключ заряда;
           ps6 ut     ;                1-й разряд -
           ps0 rapm   ;                    - " 1 " - включен
           p0B        ;                   ключ разряда;
           ps0 raps   ;                3-й разряд - вход
           p0A        ;                   тумблера "END".
           sti        ;
;;##### режим "разряд" ####################################
unload:    ps1 nul      ; обнуление часов;
           p12          ;
           p13          ;
           ld3        ; в порт 2 : режим "разряд";
           m0p pupr     ;
h0:         hlt          ; ожидание прерывания от таймера.
           bc1 end    ; >15 часов - ошибка -> конец работы.
           bvt load     ; проверка выхода компаратора.
           brn h0     ;   переход на ожидание.
;;#####  режим "заряд" ###################################
load:      p20        ;
           ld4        ;
           mr0         ;  выдача на индикатор:
           ro0        ;  сколько часов происходил
           lr0        ;  разряд, использование таблицы.
           m0p ind1   ;
           lr1        ;
           m0p ind2   ;
           ps1 nul      ; обнуление часов;
           p12          ;
           p13          ;
           ld0        ; порт 2 :"заряд";
           m0p pupr   ;
```

```
h1:       hlt         ;ожидание прерывания от ТГТ.
          bc0 h1     ; если " заряд " < 15 часов.
          l0p pupr     ; проверка состояния тумблера"END",
          cd8         ;  если включен - идти на
          bz1 end      ;  программу "КОНЕЦ",
          brn unload   ;  иначе - режим "разряд".
;;###### таблица ##########################################
tabl:  0DE       ;      порты 0  и 1 - на индикатор
       082       ;           cbga fed0 - сегменты
       0EC       ;     "2" =  1110 1100 = EC;
       0E6       ;      cbga fed0, а не abcdefg -
        0B2      ;       - по топологическим соображениям !
        076      ;
        07E      ;
        0C2      ;           bbbbb
         0FE     ;          a     c      ТАБЛИЦА
         0F6     ;          a     c     КОДИРОВКИ
         0FA     ;          a     c    7-ми СЕГМЕН-
         03E     ;           ggggg   ТНОГО ИНДИКАТОРА.
          05C    ;          f     d
          0AE    ;           f     d
          07C    ;          f     d
          078    ;          eeeee    t
;;###### подпрограмма обработки прерываний и WatchDog #####
interr:    ld1         ; Watchdog - выдача импульса обнуле-
          m0p wdog     ;  ния конденсатора CW через каждые
          ld0          ;  0.87 сек., поэтому RW*CW > 2 сек.
          m0p wdog     ;
          stc     ; 1 -> "C" -  эта единица прибавляется !
          p30       ; при ТГТ = "1" -
           sbrn reg;  - счет времени в P0;
          p03       ;  хранение времени в парах регистров:
          bc0 s     ;    P2 - старший, P3 - младший.
          p20       ; Fosc=35kHz/(4*30*256)=1.14Hz (timer)
           sbrn reg;  1.14Hz->T=0.87c*256=225сек (P3)
          p02       ;   225сек*256=57500сек=16час (P2)
s:         tg0
          ret
;;#### подпрограмма P0 + 1 -> P0 8 разр. ################
reg:       lr0 ; R0 -> младший разряд.
          ar2 ; R1 -> старший.
          mr0
          bc0 ss
          lr1
          ar2
          mr1
ss:        ret
;;#### программа "КОНЕЦ" ################################
0076:     ; пример определения метки: здесь 0076 = end !
end:       ldB       ; выдача на индикатор "H" - признак
```

```
        m0p 00   ; конца работы.
        m0p 01   ;
        ld1      ; порт 2 : "конец";
        m0p 02   ;
        hlt      ;
```

Исполняемый файл, полученный в результате трансляции этой программы ассемблером:

```
;# PRI.ASS   3D-0021-46   F6-0049-70   34-006B-102
   65-0000-19   00-0022-46   FA-004A-71   44#006C-104
   46-0001-20   71-0023-47   3E-004B-72    . . .  *)
   AF-0002-21   3D-0024-48   5C-004C-73   44#0075-104
   FF-0003-21   01-0025-48   AE-004D-74   FB-0076-105
   75-0004-22   A9-0026-49   7C-004E-75   3D-0077-106
   AE-0005-23   00-0027-49   78-004F-76   00-0078-106
   A7-0006-23   A2-0028-50   F1-0050-78   3D-0079-107
   A8-0007-24   A3-0029-51   3D-0051-79   01-007A-107
   50-0008-24   F0-002A-52   03-0052-79   F1-007B-108
   8B-0009-25   3D-002B-53   F0-0053-80   3D-007C-109
   A8-000A-26   02-002C-53   3D-0054-81   02-007D-109
   00-000B-26   45-002D-54   03-0055-81   45-007E-110
   8A-000C-27   11-002E-55   57-0056-82   ;# name # adr #
   54-000D-28   2D-002F-55   93-0057-83   ; END      0076
   A9-000E-30   7D-0030-56   18-0058-84   ; H0       0015
   00-000F-30   02-0031-56   63-0059-84   ; H1       002D
   A2-0010-31   C8-0032-57   83-005A-85   ; INIT     0000
   A3-0011-32   14-0033-58   11-005B-86   ; INTERR   0050
   F3-0012-33   76-0034-58   61-005C-86   ; IND1     0000
   3D-0013-34   10-0035-59   92-005D-87   ; IND2     0001
   02-0014-34   0E-0036-59   18-005E-88   ; LOAD     001C
   45-0015-35   44#0037-61   63-005F-88   ; NUL      0000
   12-0016-36    . . .  *)   82-0060-89   ; PUPR     0002
   76-0017-36   44#003F-61   75-0061-90   ; RAPS     0000
   16-0018-37   DE-0040-61   34-0062-91   ; RAPM     0050
   1C-0019-37   82-0041-62   70-0063-95   ; REG      0063
   10-001A-39   EC-0042-63   62-0064-96   ; S        0061
   15-001B-39   E6-0043-64   30-0065-97   ; SS       006B
   92-001C-41   B2-0044-65   11-0066-98   ; TI       00FF
   F4-001D-42   76-0045-66   6B-0067-98   ; TABL     0040
   30-001E-43   7E-0046-67   71-0068-99   ; UT       00A7
   B8-001F-44   C2-0047-68   62-0069-100  ; UNLOAD   000E
   70-0020-45   FE-0048-69   31-006A-101  ; WDOG     0003
```

*) здесь # - признак заполнения ПЗУ ассемблером.

Приложение 4. Принципиальная электрическая схема