

КОДИРОВАНИЕ ЭСППЗУ МИКРОКОНТРОЛЛЕРА

Кодирование 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 V. Кодирование микроконтроллера осуществляется с использованием выводов микроконтроллера **PB0**, **PB1** и **PB2**.

Для кодирования ЭСППЗУ может быть использован простейший программатор, схема которого приведена на рис. 19 (его по силам собрать любому претерпевшему).

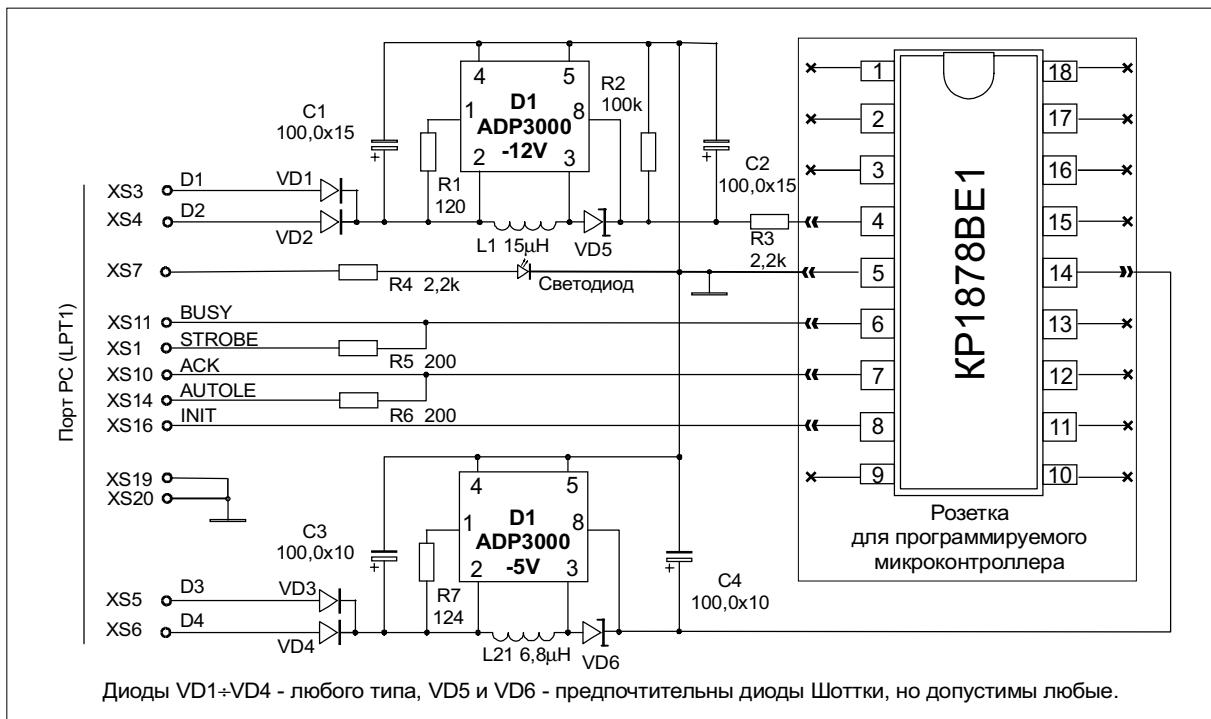


Рис. 19. Схема программатора для микроконтроллера KP1878BE1

лю микроконтроллеров) или выпускаемый партнером Ангстрема (ООО "Телесистемы", <http://www.ts/aha/ru>) промышленный программатор микроконтроллеров и E(E)PROM **PICPROG**, который программирует все контроллеры ТЕСЕЙ и более 500 типов других микросхем с числом выводов от 8 до 40.

В отличие от многих других программаторов, многие из которых также могут быть использованы, **PICPROG** позволяет программировать и верифицировать микросхемы при различных напряжениях питания, гарантируя надежность программирования.

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 данных. Чтение ее содержимого можно осуществить только после записи в нее информации в пределах одного цикла программирования. Таким образом, полностью гарантируется конфиденциальность запрограммированной управляющей программы.

Строка конфигурации определяет режимы работы генератора тактовой частоты ми-



крооконтроллера и включение счетчика задержки начального пуска процессора. Следует отметить, что при программировании строки конфигурации каждый ее разряд необходимо повторять три раза подряд. Стока конфигурации при программировании рассматривается как 1025-е слово ЭСППЗУ команд.

Формат строки конфигурации

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

RDE – разряд включения счетчика задержки начального пуска процессора. Если **RDE = 1**, счетчик задержки начального пуска процессора включен, если **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 kHz до 8 MHz;

OT1 = 1, OT2 = 1 – режим генерации тактовой частоты с использованием внешнего кварцевого резонатора с частотами до 500 kHz, а также режим трансляции внешней тактовой частоты;

OT1 = 0, OT2 = 1 – режим подключения внутреннего генератора тактовой частоты » 50 kHz.

Далее приводится последовательность действий программатора и микроконтроллера во время одного цикла программирования микроконтроллера, в котором записываются ЭСППЗУ команд, строка конфигурации и ЭСППЗУ данных, и производится контроль правильности этой записи.

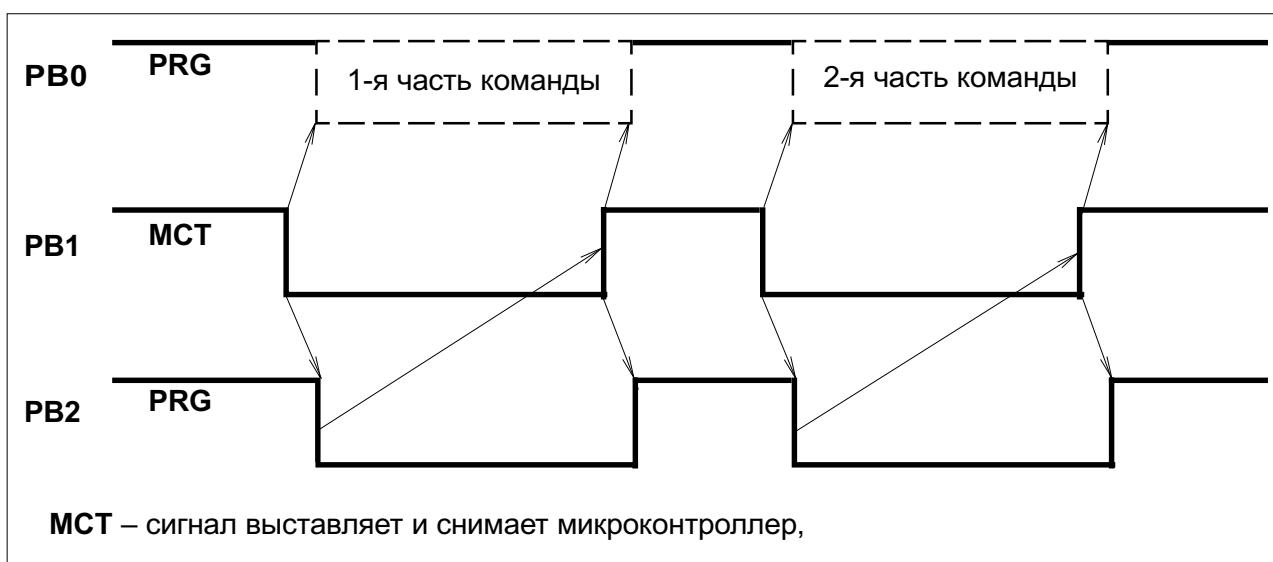


Рис. 20. Временные диаграммы процедуры приема микроконтроллером команды от программатора



ПРИЕМ МИКРОКОНТРОЛЛЕРОМ КОМАНДЫ ОТ ПРОГРАММАТОРА

МИКРОКОНТРОЛЛЕР

ПРОГРАММАТОР

2. После подачи на вывод **RST** уровня напряжения +12 V, на выводах **PB0**, **PB1** и **PB2** включаются нагрузочные резисторы, обеспечивающие высокий уровень напряжения на этих выводах.

Производится обнуление ЭСППЗУ команд и последующая проверка правильности этой операции. Если обнуления ЭСППЗУ команд не произошло, микроконтроллер завершает работу в режиме программирования.

Если контроль ЭСППЗУ команд на обнуление завершился успешно, на вывод **PB1** выставляется низкий уровень, означающий, что микроконтроллер готов к приему команды.

4. Микроконтроллер, получив низкий уровень на выводе **PB2**, подтверждает получение первой части команды снятием низкого уровня с вывода **PB1** и ждет высокого уровня на выводе **PB2**.

6. При установке на выводе **PB2** высокого уровня на вывод **PB1** выставляется низкий уровень, означающий, что микроконтроллер готов к приему второй части команды.

8. Микроконтроллер, получив низкий уровень на выводе **PB2**, подтверждает получение второй части команды снятием низкого уровня с вывода **PB1** и ждет высокого уровня на выводе **PB2**, чтобы приступить к выполнению команды.

1. Программатор подает на вывод **RST** микроконтроллера напряжение +12 V для установки режима программирования.

3. Программатор контролирует состояние выводов **PB0**, **PB1** и **PB2**. Если на них установленся высокий уровень, то ожидается (через £10s) появление низкого уровня на выводе **PB1**, означающего, что микроконтроллер готов к приему команды. Если низкий уровень не появился, это признак ошибочного состояния при программировании.

После получения готовности микроконтроллера программатор подает на вывод **PB0** первую часть команды обмена. Высокий уровень на **PB0** означает, что будет вестись работа с ЭСППЗУ команд, а низкий – работа с ЭСППЗУ данных. Первая часть команды подтверждается программатором высвятилением низкого уровня на вывод **PB2**.

Программатор переходит к ожиданию снятия микроконтроллером низкого уровня с вывода **PB1**, означающего, что микроконтроллер принял первую часть команды.

5. После снятия микроконтроллером низкого уровня с вывода **PB1** программатор должен снять низкий уровень с вывода **PB2**, установить высокий уровень на выводе **PB0** и ожидать готовности микроконтроллера к приему второй части команды.

7. После получения готовности микроконтроллера программатор подает на вывод **PB0** вторую часть команды обмена. Высокий уровень на **PB0** означает, что будет происходить запись, а низкий – чтение. Вторая часть команды подтверждается программатором высвятилением низкого уровня на вывод **PB2**. Программатор переходит к ожиданию снятия микроконтроллером низкого уровня с вывода **PB1**, означающего, что микроконтроллер принял вторую часть команды.

9. После снятия микроконтроллером низкого уровня с вывода **PB1** программатор должен снять низкий уровень с вывода **PB2**, установить высокий уровень на выводе **PB0** и ожидать готовности микроконтроллера к выполнению требуемой команды.



ЗАПИСЬ ПРОГРАММАТОРОМ ДАННЫХ В ПАМЯТЬ МИКРОКОНТРОЛЛЕРА

МИКРОКОНТРОЛЛЕР

ПРОГРАММАТОР

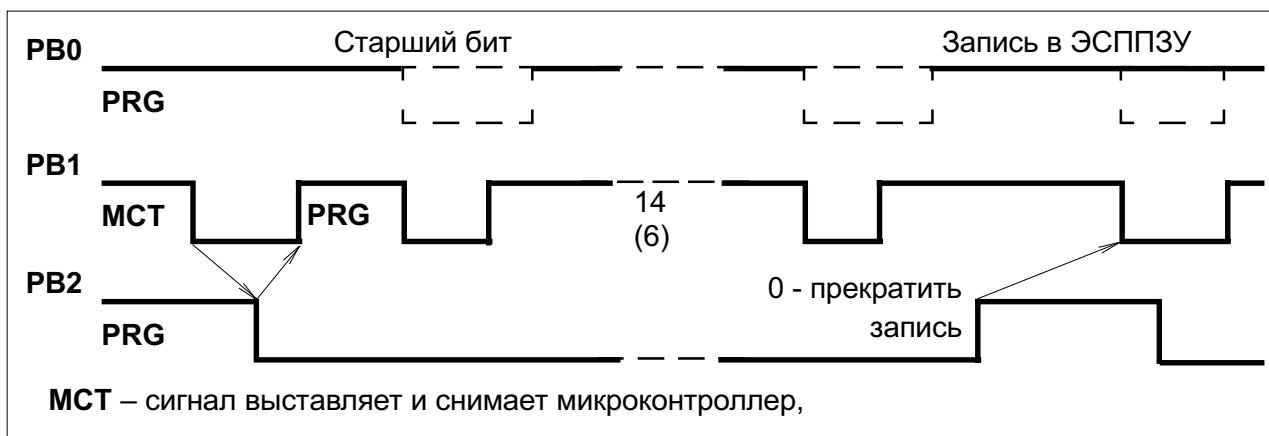
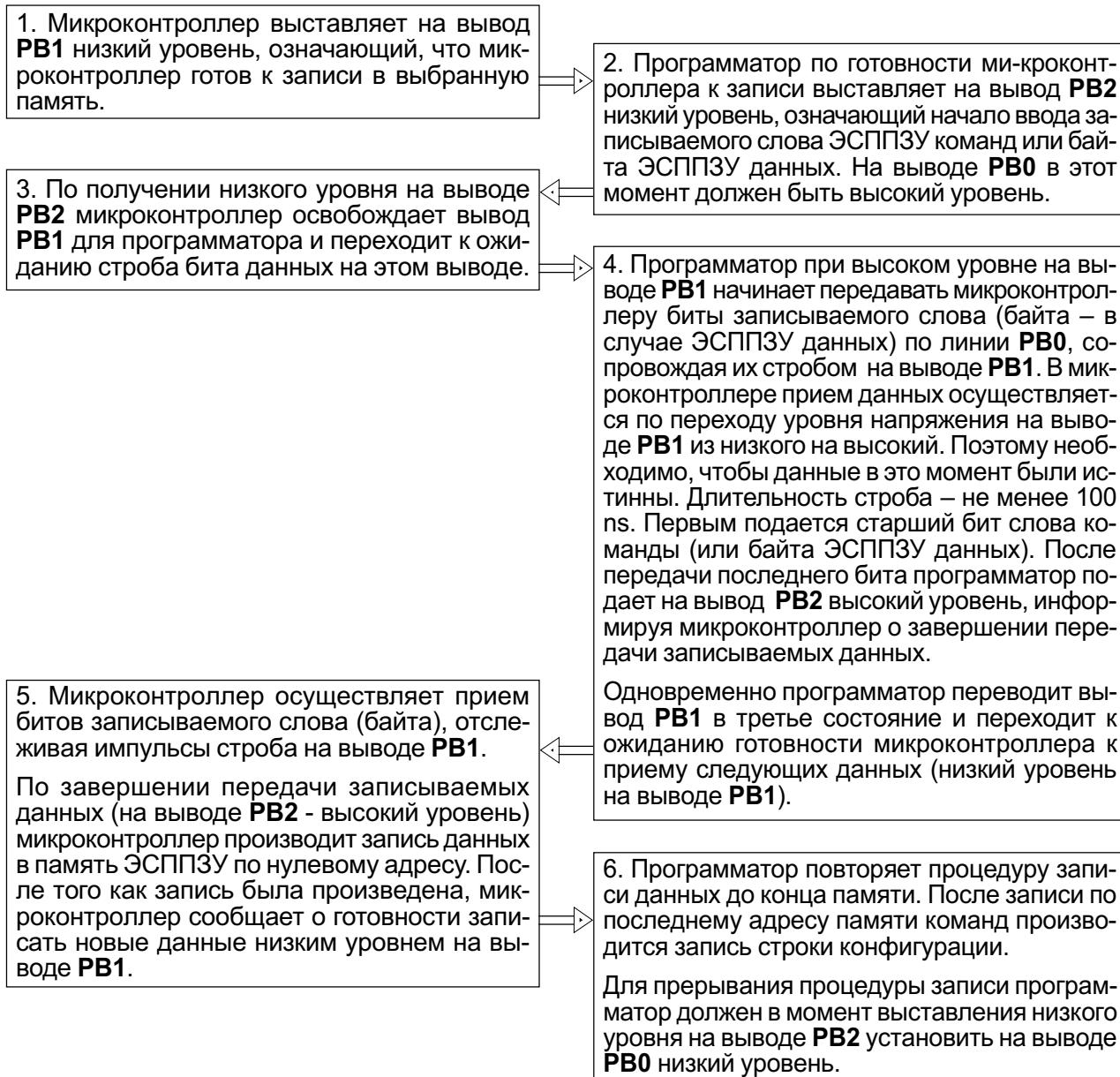


Рис. 21. Временные диаграммы процедуры записи программатором данных в память ЭСППЗУ микроконтроллера.



ЧТЕНИЕ ПРОГРАММАТОРОМ ДАННЫХ ИЗ ПАМЯТИ МИКРОКОНТРОЛЛЕРА

МИКРОКОНТРОЛЛЕР

ПРОГРАММАТОР

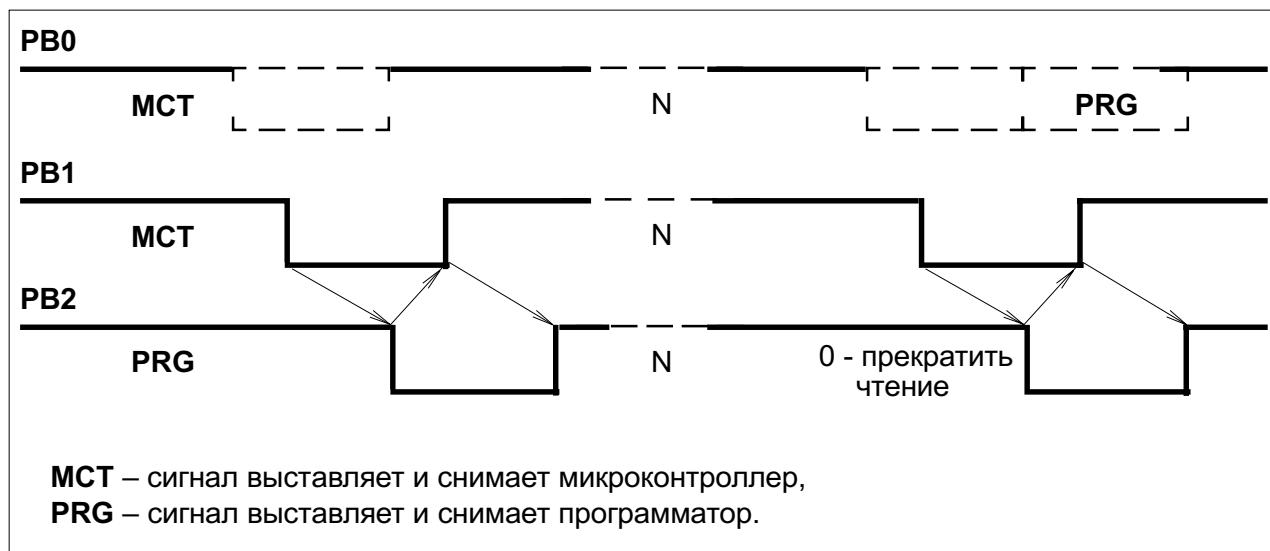
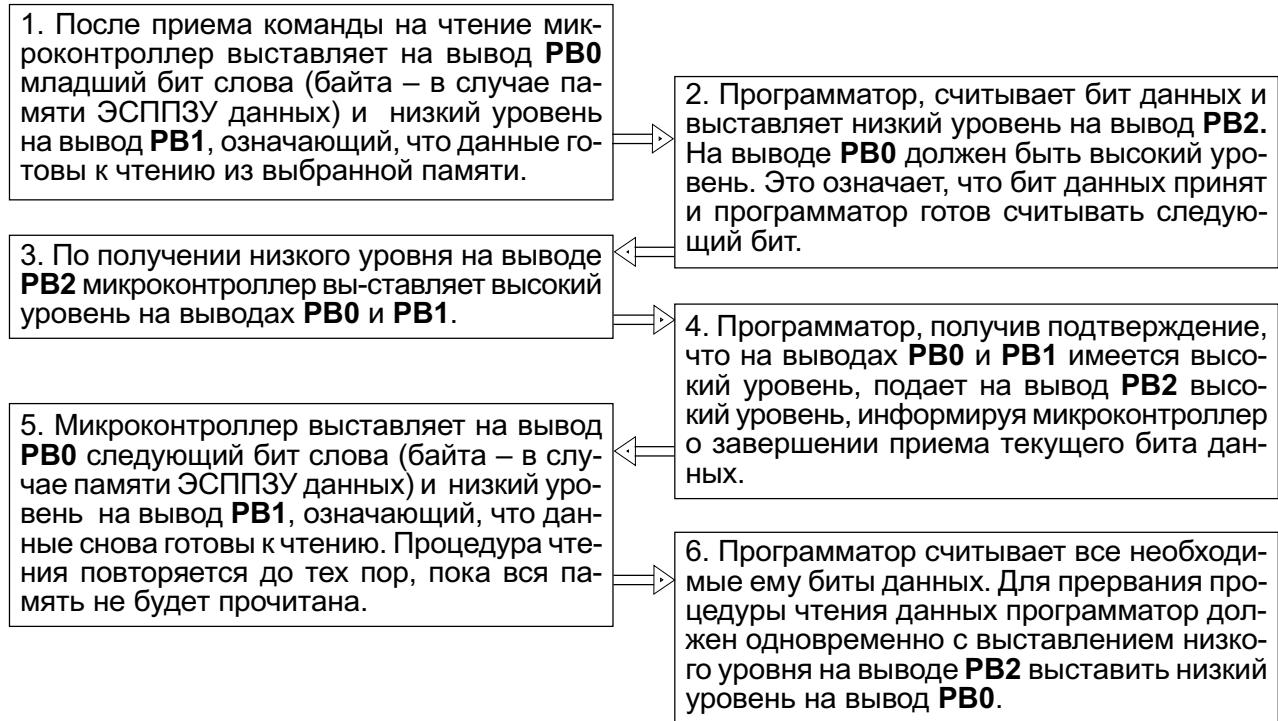


Рис. 22. Временные диаграммы процедуры чтения программатором данных из памяти ЭСППЗУ микроконтроллера.