Спецификация по программированию и тестированию внутренней памяти программ микроконтроллеров 1886ВЕ1У, 1886ВЕ2У, 1886ВЕ3У и 1886ВЕ4У.

1. Подключение микроконтроллера.

Для программирования и тестирования внутренней памяти программ микроконтроллеров 1886ВЕ1У, 1886ВЕ2У, 1886ВЕ3У и 1886ВЕ4У используется последовательный интерфейс ISP (Interface Serial Program). В этом режиме программирования задействованы семь выводов микроконтроллера, приведённые в таблице 1. Примечание: вывод RX/DT должен иметь «подтяжку» к напряжению питания U_{DD}.

Обозначение	В режиме программирования		
	Назначение	Тип	Описание
PA4/RX/DT	DT	I/O	Последовательные данные
PA5/TX/CK	CK	I	Последовательный синхросигнал
PA1/T0CKI	OSCI	I	Источник синхронизации микроконтроллера
TEST	TEST	I	Вход для выбора тестового режима
nMCLR/U_{PP}	nMCLR/U_{PP}	P	Напряжение программирования
U_{DD}	U_{DD}	P	Напряжение питания
U_{SS}	U_{SS}	P	Цифровая «земля»

Таблица 1А. Выводы ISP интерфейса для микроконтроллеров 1886ВЕ1У и 1886ВЕ2У.

Обозначение	В режиме программирования		
	Назначение	Тип	Описание
PA2/RX/DT	DT	I/O	Последовательные данные
PA3/TX/CK	CK	I	Последовательный синхросигнал
PA1/T0CKI	OSCI	I	Источник синхронизации микроконтроллера
TEST	TEST	I	Вход для выбора тестового режима
nMCLR/U_{PP}	nMCLR/U_{PP}	P	Напряжение программирования
U_{DD}	U_{DD}	P	Напряжение питания
U_{SS}	U_{SS}	P	Цифровая «земля»

Таблица 1Б. Выводы ISP интерфейса для микроконтроллеров 1886ВЕ3У и 1886ВЕ4У.

2. Перевод микроконтроллера в режим последовательного программирования.

Перевод микроконтроллера в режим последовательного программирования (ISP) осуществляется подачей напряжения программирования V_{IHH} на выводы TEST и nMCLR/U_{PP} в следующей последовательности:

1. Подача V_{IHH} на вывод TEST.

2. Подача V_{IHH} на вывод nMCLR/U_{PP}.

Время установки между событиями 1 и 2 должно быть не менее 1 микросекунды. В итоге счётчик команд будет указывать на адрес памяти программ 0xFF00. В этой области памяти программ расположено загрузочное ПЗУ (Boot ROM). Этот программный код инициализирует USART/SCI для получения команд. Для выполнения программы из загрузочного ПЗУ необходимо обеспечить подачу синхросигнала на вывод PA1/T0CKI. Для того, чтобы USART/SCI был инициализирован для получения команд, необходимо выполнить следующую последовательность действий:

1. Запустить источник синхронизации микроконтроллера.

2. Ожидать 300 000 циклов микроконтроллера для инициализации USART/SCI загрузочным ПЗУ. Для частоты синхронизации 32 МГц – пауза 37.5 мс.

3. Послать команду по USART/SCI.

3. Команды управления программированием.

В ISP режиме модуль USART/SCI сконфигурирован как синхронное ведомое устройство. Микроконтроллер ожидает команду, которая выполняется загрузочным ПЗУ, расположенным в области программной памяти 0xFF00-0xFFFF. Программно-аппаратные средства ISP поддерживают 20 команд, приведённых в таблице 2. Микроконтроллер 1886ВЕ1У имеет память программ масочного типа, поэтому команды записи/стирания памяти для него не доступны.

команда	**код команды**	**пара-метры**	**ответ**	**приме-чание**
СБРОС УКАЗАТЕЛЯ ПАМЯТИ ПРОГРАММ	0000 0000	-	-	-
ЗАГРУЗКА ДАННЫХ	0000 0010	2 байта		
ЧТЕНИЕ ДАННЫХ	0000 0100		2 байта	
ИНКРЕМЕНТ АДРЕСА	0000 0110			
СТАРТ ПРОГРАММИРОВАНИЯ	0000 1000			5
ЗАГРУЗКА АДРЕСА	0000 1010	2 байта		
ЧТЕНИЕ АДРЕСА	0000 1100		2 байта	
СТОП ПРОГРАММИРОВАНИЯ	0000 1110			5
ЗАПИСЬ ДЕЛИТЕЛЯ ЧАСТОТЫ	0000 0001	2 байта		3
Зарезервировано	0000 0011		2 байта	
Зарезервировано	0000 0111			
СТИРАНИЕ СЕКТОРА	0000 1111	2 байта		3, 5
ТЕСТ СЕКТОРА НА ЧИСТОТУ	0000 0101	2 байта	2 байта	3, 5
Расширение стандартного набора команд				
ЗАПИСЬ ВСЕЙ ПАМЯТИ ЗНАЧЕНИЕМ 0FFFFh	0000 1001			1, 2, 4, 5
ТЕСТ ВСЕЙ ПАМЯТИ НА ЗНАЧЕНИЕ 0FFFFh	0000 1011		1 байт	1, 5
ЧТЕНИЕ ОЗУ + ПЕРИФЕРИЯ	0001 0000	2 байта	1 байт	
ЗАПИСЬ ОЗУ + ПЕРИФЕРИЯ	0001 0001	3 байта		
ЗАПРОС ГОТОВНОСТИ	0001 0010		1 байт	1
ПЕРЕХОД GOTO	0001 0100	2 байта		6
ВЕРСИЯ МИКРОКОНТРОЛЛЕРА	0001 0101		1 байт	1

Таблица 2. Команды ISP.

Примечания:

1. Команда работает в «защищенном» режиме микроконтроллера.
2. После выполнения команды должна производиться синхронизация с помощью команды «запрос готовности».

3. Команда работает в «защищенном» режиме в случае успешного завершения выполнения команд «запись всей памяти значением 0FFFFh» или «тест всей памяти на значение 0FFFFh».
4. Для микроконтроллеров 1886ВЕ3У и 1886ВЕ4У производится стирание внутренней EEPROM памяти данных.
5. Для микроконтроллера 1886ВЕ1У указанные команды не поддерживаются. Для совместимости с другими микроконтроллерами, микроконтроллер производит выборку параметров этих команд и выдает ответ (ответ содержит не достоверную информацию).
6. Для микроконтроллера 1886ВЕ1У команда «ПЕРЕХОД GOTO» работает в «защищенном» режиме микроконтроллера при адресации в область памяти 7F00h-7FEFh.

3.1 Сброс указателя памяти программ.

Эта команда предназначена для очистки указателя адреса памяти программ. После этого указатель находиться в исходном состоянии и указывает на адрес 0x0000. Временная диаграмма команды приведена на рисунке 1.

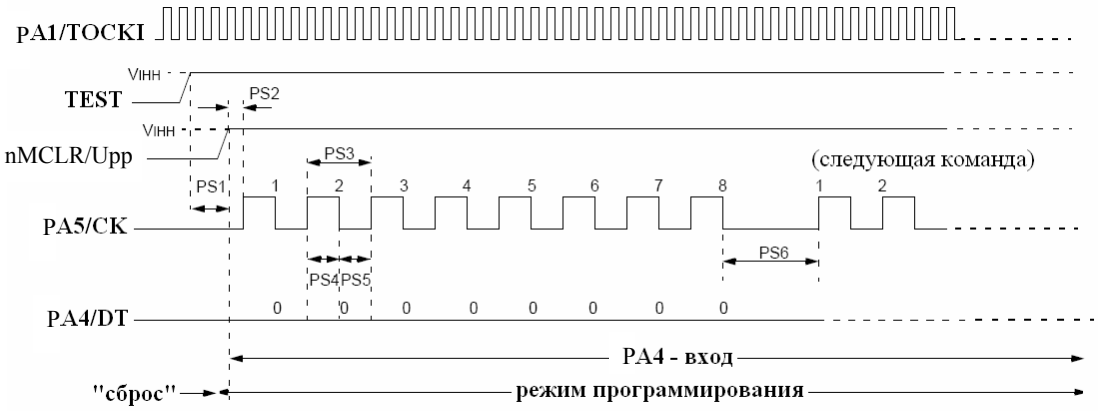

Рисунок 1. Команда сброса указателя адреса памяти программ.

3.2 Инкремент адреса.

Эта команда инкрементирует указатель адреса памяти программ. Обычно эта команда следует за командами чтения или программирования. Временная диаграмма команды приведена на рисунке 2.

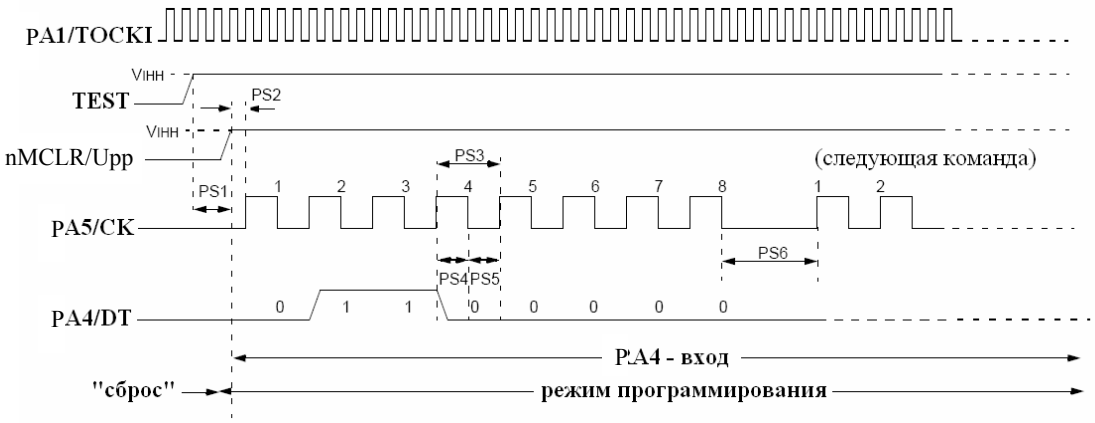

Рисунок 2. Команда инкремента адреса.

3.3 Загрузка адреса.

Эта команда используется для загрузки указателя адреса программ определённым 16 разрядным значением. Обычно эта команда применяется для доступа к определённому адресному пространству. Временная диаграмма команды приведена на рисунке 3.

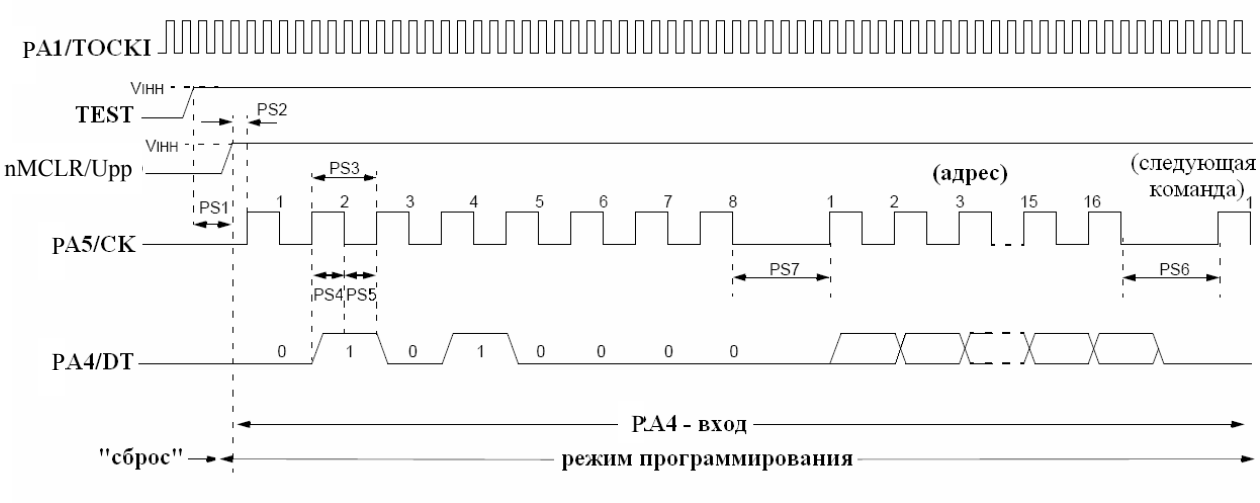

Рисунок 3. Команда загрузки адреса памяти программ.

3.4 Чтение адреса.

Эта команда используется для получения текущего указателя адреса памяти программ. Временная диаграмма команды приведена на рисунке 4.

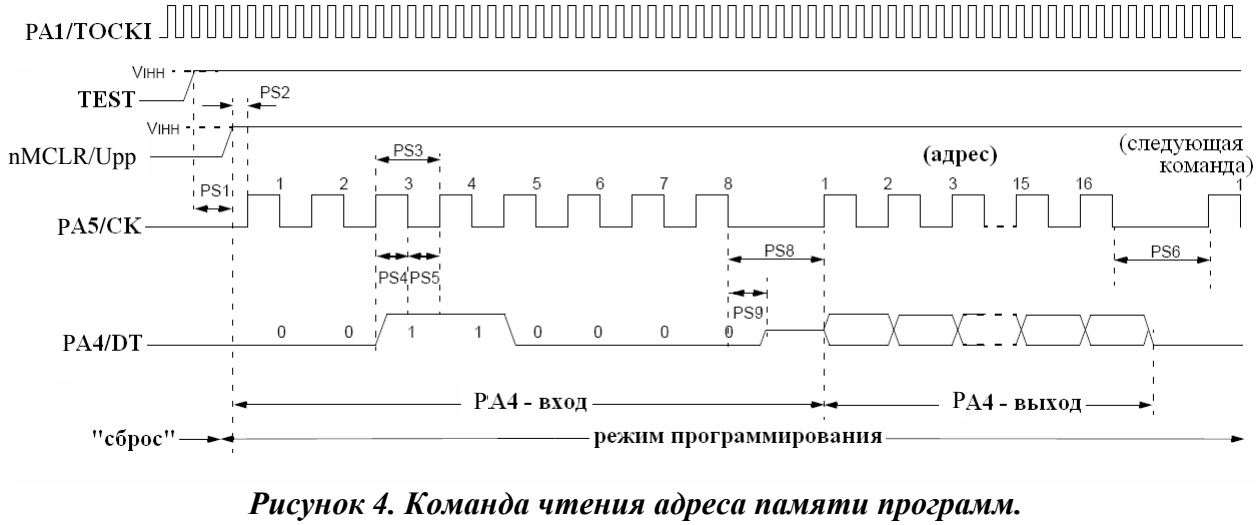

Рисунок 4. Команда чтения адреса памяти программ.

3.5 Загрузка данных.

Эта команда загружает 16 разрядные данные, которые должны быть запрограммированы в память программ. Адрес памяти программ может быть модифицирован после загрузки данных. Данные не будут запрограммированы, пока не

выполниться команда СТАРТ ПРОГРАММИРОВАНИЯ. Временная диаграмма команды приведена на рисунке 5.

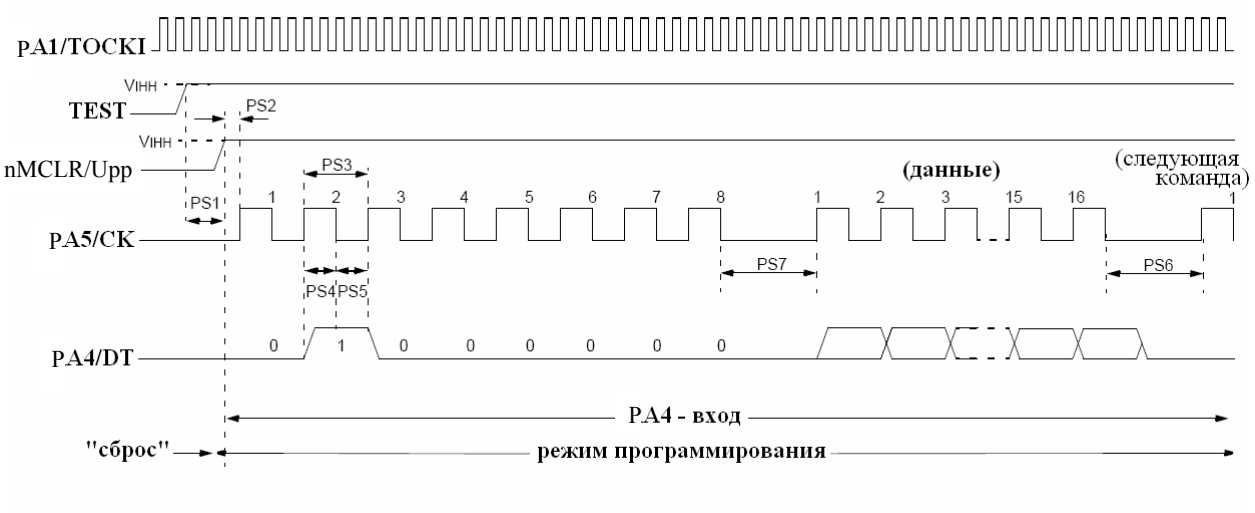

Рисунок 5. Команда загрузки данных.

3.6 Чтение данных.

Эта команда читает данные, расположенные по адресу на который ссылается указатель адреса памяти программ. Обычно эта процедура используется в процедуре контроля правильности запрограммированных данных. Временная диаграмма команды приведена на рисунке 6.

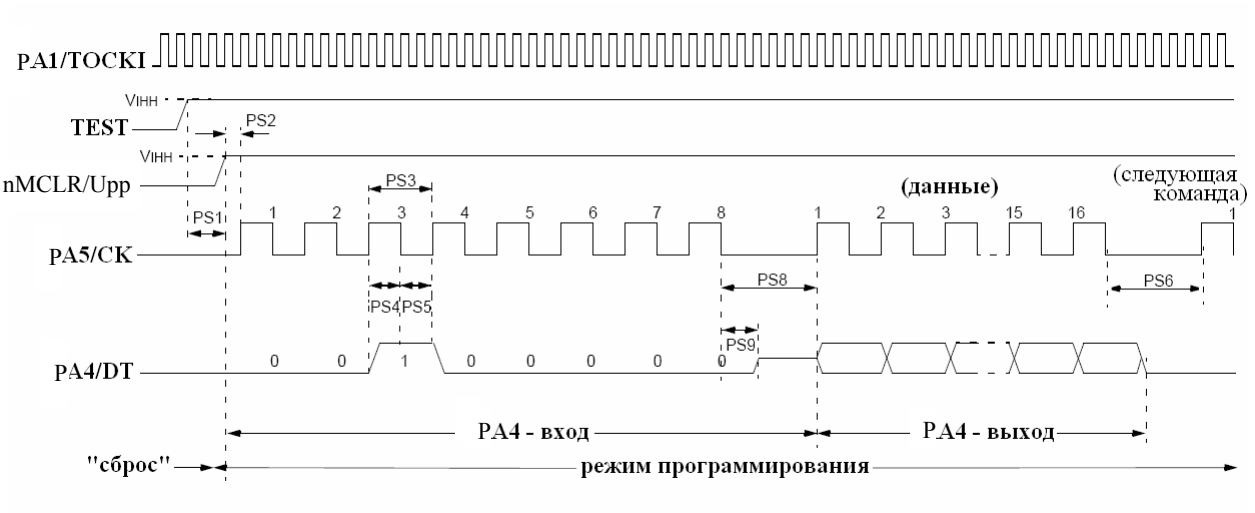

Рисунок 6. Команда чтения данных.

3.7 Старт программирования.

Эта команда используется для программирования текущих данных (посланных последней командой ЗАГРУЗКА ДАННЫХ) по адресу памяти программ на который ссылается указатель адреса. Время программирования данных 200 микросекунд. После прохождения этого времени любая команда может быть послана и выполнена. Временная диаграмма команды приведена на рисунке 7.

Примечание. С целью увеличения надежностных характеристик программирование памяти рекомендуется проводить дважды.

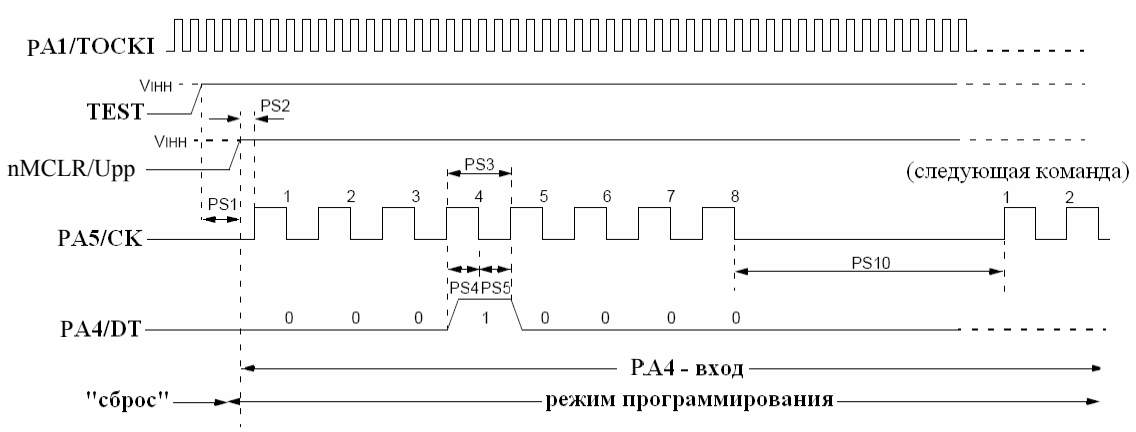

Рисунок 7. Команда программирования.

3.8 Запись делителя частоты.

Эта команда предназначена для записи коэффициента деления частоты генератора в 8 разрядный регистр микроконтроллера (WRDIV) с целью выработки импульса WR (запись во флэш) длительностью 5 мкс. Делитель передается в первом байте 16 разрядных данных. Правило выбора коэффициента деления:

K = 5 мкс / Tosc (Tosc в мкс).

В случае если предполагается записывать или стирать память и частота генератора отличается от 1 МГц данный регистр необходимо предварительно установить в нужное состояние согласно правилу выбора коэффициента деления. Временная диаграмма команды приведена на рисунке 8.

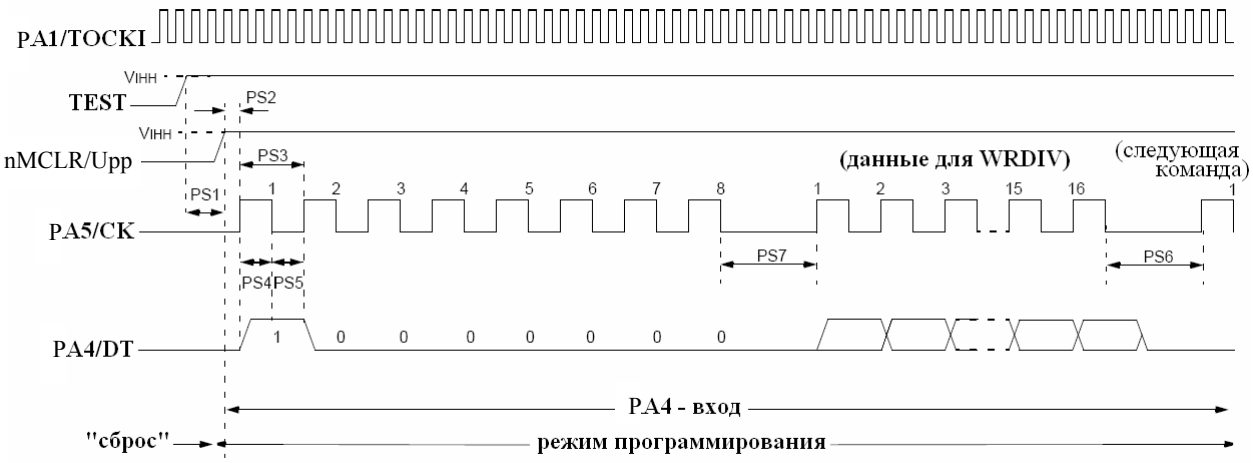

Рисунок 8. Команда записи делителя частоты.

3.9 Стирание сектора.

Эта команда предназначена для запуска процедуры стирания одного из 4 секторов флэш-памяти. Стираемый сектор выбирается 16 разрядным адресом начала сектора, передаваемым в команде:
1 сектор – 0x0000;
2 сектор – 0x2000;
3 сектор – 0x4000;
4 сектор – 0x6000.
Первым передается старший байт начального адреса стираемого сектора.
Процедура Depletion test проводиться только после успешного стирания двух последовательных секторов. Временная диаграмма команды приведена на рисунке 10.

Примечания.
1. После успешно проведенного стирания (проверка сектора на чистоту проведена успешно) необходимо повторить процедуру стирания 2 или 3 раза, с целью повышения надежности.
2. Проверка сектора на чистоту (при контроле стирания) должна проводиться при предполагаемой максимальной рабочей частоте микроконтроллера и напряжении питания 4.5 вольта.

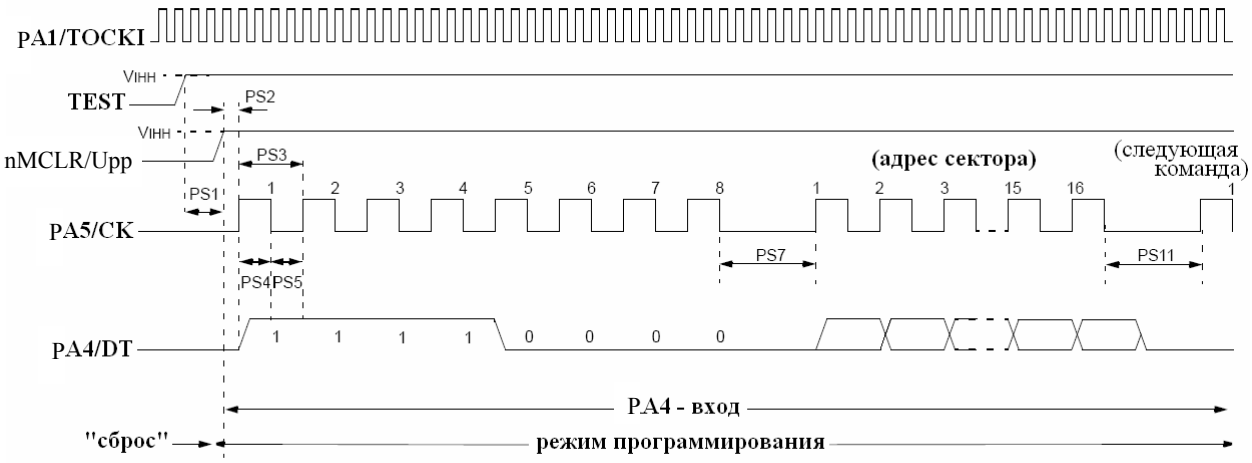

Рисунок 10. Команда стирания сектора .

3.10 Тест сектора на чистоту.

Эта процедура проводиться после операции стирание сектора, для проверки чистоты сектора. Проверяемый сектор выбирается 16 разрядным адресом начала сектора, передаваемым в команде:
1 сектор – 0x0000;
2 сектор – 0x2000;
3 сектор – 0x4000;
4 сектор – 0x6000.
В случае, если все ячейки памяти в адресном пространстве проверяемого сектора содержат значение 0x0000, сектор считается очищенным и из канала USART/SCI

читаются 16 разрядные данные 0xFFFF, в противном случае читаются 0x0000. Временная диаграмма команды приведена на рисунке 12.

Примечание. Тест сектора на чистоту должен проводиться при предполагаемой максимальной рабочей частоте микроконтроллера и напряжении питания 4.5 вольта.

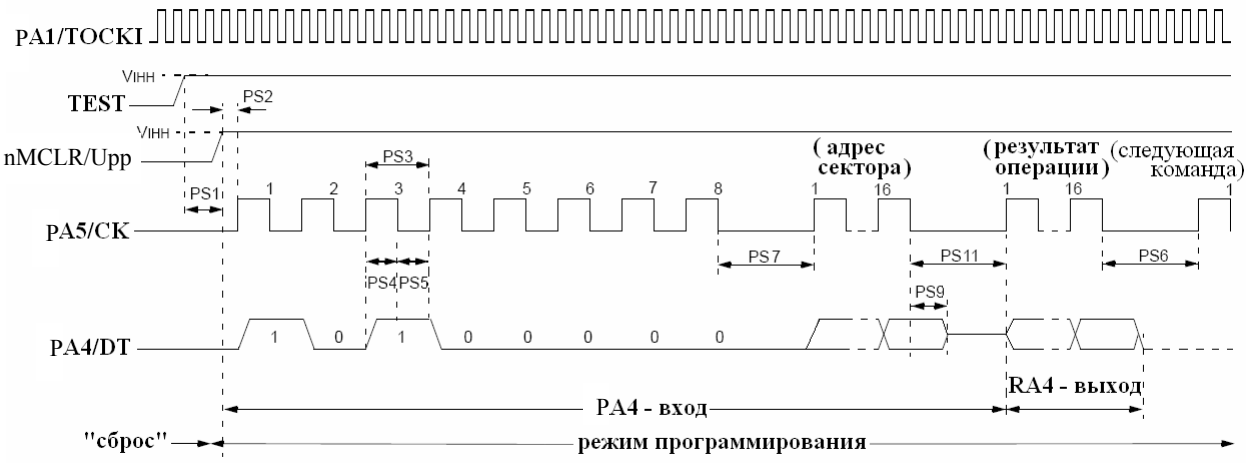

Рисунок 12. Команда тест сектора на чистоту.

3.11 Запись всей памяти значением 0FFFFh.

Эта процедура применяется для вывода микроконтроллера из «защищенного» режима. После выполнения процедуры вся память микроконтроллера принимает гарантированное значение равное 0FFFFh, блокируя возможность восстановления ранее записанной информации. Для микроконтроллеров 1886ВЕ3У и 1886ВЕ4У дополнительно производится стирание внутренней EEPROM памяти данных. Успешное выполнение процедуры разрешает запуск процедур «стирание сектора», «тест сектора на чистоту» и «запись делителя частоты» в «защищенном» режиме микроконтроллера. После выполнения процедуры необходимо произвести синхронизацию с помощью команды «запрос готовности». Для запуска процедуры передается один байт с кодом 09h.

3.12 Тест всей памяти на значение 0FFFFh.

Эта процедура применяется для сравнения значения содержимого памяти с 0FFFFh и выдачи информации о результате сравнения. Процедура может быть запущена в «защищенном» режиме микроконтроллера. В случае совпадения значения памяти с 0FFFFh, разрешается запуск процедур «стирание сектора», «тест сектора на чистоту» и «запись делителя частоты» в «защищенном» режиме микроконтроллера. Для запуска процедуры передается один байт с кодом 0Bh, отсчитывается пауза не менее 400000*T_{CY} (для частоты синхронизации 32 МГц – 50 мс) и принимается один байт ответа. В случае совпадения значения памяти с 0FFFFh выдается ответ 55h, в случае не совпадения 0FFh.

3.13 Чтение ОЗУ + периферия.

Эта процедура применяется для чтения содержимого периферийных регистров или ОЗУ программируемого микроконтроллера. Для запуска процедуры последовательно

передаются три байта: код (10h), банк ОЗУ (регистр BSR) и адрес считываемого регистра, и далее принимается байт со считанным значением.

3.14 Запись ОЗУ + периферия.

Эта процедура применяется для записи периферийных регистров или ОЗУ программируемого микроконтроллера. Для запуска процедуры последовательно передаются четыре байта: код (11h), банк ОЗУ (регистр BSR), адрес записываемого регистра и записываемое значение.

3.15 Запрос готовности.

Процедура применяется для определения готовности микроконтроллера, а также для синхронизации обмена командами в случае выполнения процедур с не фиксированной длительностью исполнения, например: «запись всей памяти значением 0FFFFh». Для запуска процедуры передается один байт с кодом 12h и считывается один байт ответа. Если принятое значение отличается от 55h, то микроконтроллер не готов и синхронизация не восстановлена (ожидаемое значение при неготовности микроконтроллера 0FFh). Далее производится повтор процедуры. После получения ответа равного 55h микроконтроллер готов к обработке следующих процедур.

Процедура может быть запущена в «защищенном» режиме микроконтроллера.

3.16 Переход GOTO.

Эта процедура применяется для запуска программы в памяти микроконтроллера. Для запуска процедуры последовательно передаются три байта: код (14h), младшая половина адреса и старшая половина адреса.

В «защищенном» режиме микроконтроллера процедура может быть выполнена только для микроконтроллера 1886ВЕ1У при адресации в область памяти 7F00h-7FEFh.

3.17 Версия микроконтроллера.

Эта процедура возвращает номер версии микроконтроллера. Для запуска процедуры передается байт с кодом 15h и считывается байт с номером версии. Процедура может быть запущена в «защищенном» режиме микроконтроллера. Для микроконтроллера 1886ВЕ1У значение байта X1h, для 1886ВЕ2У – X2h, для 1886ВЕ3У – X3h, для 1886ВЕ4У – X4h.

3.18 Доступ к FLASH памяти в различных режимах микроконтроллера.

Микроконтроллеры имеют четыре режима:
- микропроцессор;
- микроконтроллер;
- расширенный микроконтроллер;
- «защищенный» режим, или режим «защищенного» микроконтроллера.

Подробнее смотрите спецификации на микроконтроллеры серии 1886.

«Защищенный» режим предназначен для защиты содержимого памяти микроконтроллера от несанкционированного считывания или записи. При установке этого режима блокируется выполнение большинства команд доступа к памяти (смотрите таблицу 2). Для вывода микроконтроллера из «защищенного» режима необходимо

произвести «уничтожение» ранее записанной информации с помощью процедуры «запись всей памяти значением 0FFFFh», и далее, не снимая питания микроконтроллера, произвести полное стирание FLASH памяти. В случае, если вся память микроконтроллера, записана значением 0FFFFh, то для разрешения стирания можно выполнить процедуру «тест всей памяти на значение 0FFFFh».

Режим микропроцессора предназначен для работы микроконтроллера только с внешней памятью программ, внутренняя FLASH память при этом отключается. Если записать информацию отличную от 0000h в память микроконтроллера и установить режим микропроцессора, то произойдет блокирование чтения/записи внутренней памяти. Выход из этого состояния аналогичен выходу из «защищенного» режима.

4. Электрические характеристики для последовательного режима программирования.

Пара-метр	Характеристика	Мин.	Тип.	Макс.	Единицы измерен.
V_{IHH}	Напряжение программирования на выводах **nMCLR/U_{PP}** и **TEST**	11.75	12	12.25	В
I_{PP}	Ток программирования на входе **nMCLR/U_{PP}**	-	25	50	мА
Fosc	Входная частота на **PA1**	1	-	33	МГц
T_{CY}	Такт инструкции микроконтроллера	-	4/Fosc	-	
PS1	Время между установкой **V**IHH на выводах **TEST** и **nMCLR/U_{PP}**	1	-	-	мкс
PS2	Время инициализации интерфейса	300000	-	-	T_{CY}
PS3	Период частоты	1	-	-	T_{CY}
PS4	Время установки данных	15	-	-	нс
PS5	Время удержания данных	15	-	-	
PS6	Задержка между последним спадом сигнала CK и первым фронтом CK следующей команды	160	-	-	T_{CY}
PS7	Задержка между последним спадом сигнала CK передачи байта команды и первым фронтом CK записываемого слова данных	160	-	-	T_{CY}
PS8	Задержка между последним спадом сигнала CK передачи байта команды и первым фронтом CK читаемого слова данных	160	-	-	T_{CY}
PS9	Время перенастройки вывода данных с выхода на вход	1	-	-	T_{CY}
PS10	Минимальное время программирования	200	-	-	мкс
PS11	Время выполнения процедуры **DPLS**, **WLS**, тестирования и стирания сектора	500	-	-	мс