

CATALIST от ON Semiconductor: качественная EEPROM без лишних материальных затрат

Ирина РОМАДИНА
romadina@compel.ru

Широта использования микросхем энергонезависимой EEPROM обуславливается известными преимуществами — высоким допустимым количеством циклов перезаписи, простотой внедрения и низкой стоимостью. Применение памяти такого вида может быть эффективно там, где необходимо запоминать относительно небольшие объемы часто обновляемой информации, а использование flash избыточно. Это хороший выбор для массовых проектов.

Введение

За предыдущие годы использования EEPROM разработчикам удалось настолько отшлифовать конструкцию и параметры данных устройств, что конкуренция в этой области осуществлялась в основном за счет маркетинговых предложений, совершенствование же характеристик происходило довольно медленно (например, только в этом году был анонсирован выпуск микросхем с плотностью 2 Мбит, и то частота работы при этом осталась на уровне 5 МГц [1]). В связи с этим рынок достаточно насыщен предложениями нескольких ведущих компаний-производителей (в частности, Microchip и Atmel) и найти там место новому бренду непросто. Однако компании ON Semiconductor, похоже, удалось это сделать: параметры их микросхем такие же, как у конкурентов. В чем же причины успеха? Ответ прост: доступность и цена.

По объективным причинам у вышеупомянутых производителей в последнее время бывали перебои с поставкой товара (в том числе и в связи с событиями в Японии — землетрясение, цунами, перебои в энергоснабжении из-за взрыва на станции Фукусима) [2]. Поэтому потребители часто вынуждены были длительно ожидать поставки необходимых микросхем или размещать долгосрочные предварительные заказы. В этих условиях как нельзя кстати приходится EEPROM производства компании Catalyst, которая с 2008 г. принадлежит ON Semiconductor. Это серия полновыводных (pin-to-pin) аналогов популярных EEPROM, причем по ценам чуть ли не в два раза ниже, чем у Atmel и Microchip, и с гораздо более краткими сроками поставки. Набор совместимых устройств, выпускаемых компанией ON Semiconductor, состоит из большого количества различных моделей микросхем, различающихся количеством па-

мяти и интерфейсом последовательной передачи данных.

Серия CAT25xxx

Серия CAT25xxx работает с интерфейсом SPI при напряжениях питания 2,5–5,5 В и тактирующих частотах до 10 МГц (при напряжении 1,8–2,5 В — до 5 МГц). В случае необходимости пользователем может быть задействована как аппаратная, так и программная защита от записи (с помощью установки соответствующих бит). Для микросхемы декларируется до миллиона циклов перезаписи и хранение данных не менее 100 лет (при нормальных температурах). Рабочий диапазон температур –40...+125 °С (микросхемы, маркированные индексом E (extended)) или –40...+85 °С (маркированные индексом I (industrial)) [3]. Следует особо отметить более низкое энергопотребление микросхем — 2 мА в режиме чтения, 4 мА в режиме записи и до 1 мкА в режиме ожидания. Аналогичные параметры в референсных схемах составляют 7 мА, 6 мА и 5 мкА соответственно. CAT25 xxx являются pin-to-pin заменой для микросхем серии AT25xxx производства Atmel и серии 25LCxxx от Microchip (рис. 1).

Из вышеизложенного понятно, что микросхема работает по интерфейсу SPI. В связи с этим назначение выводов SI (serial data input), SO (serial data output), SCK (serial clock), CS (chip select) стандартно: последовательные ввод и вывод данных, вход для так-

товой частоты от мастер-устройства (например, от микроконтроллера), выбор ведомого. Причем последний вывод также используется для коммуникации — при начале каждой передачи он устанавливается в состояние «лог. 0», после конца передачи возвращается в «лог. 1», при этом устройство автоматически переводится в режим ожидания и снижается потребление энергии. Что касается остальных выводов, то WP используется при защите от записи в регистр статуса (и только!), HOLD необходим для временной приостановки передачи данных между мастер-устройством и памятью.

Для управления устройством и проверки его состояния используется регистр статуса. Микросхема поддерживает шесть команд (кодов операций, опкодов): чтение (READ) и запись (WRITE) в память, чтение (RDSR) и запись (WRSR) в регистр статуса, установка (WREN) и сброс (WRDI) бита разрешения записи.

Общая схема каждой передачи такова: CS устанавливается в «0» (старт транзакции), передается опкод, адрес (а) ячеек и значения байтов (если необходимо), считываются результаты (если необходимо), CS устанавливается в «лог. 1» (конец транзакции). Все время, пока идет транзакция, на вывод SCK должен подаваться тактовый сигнал. Необходимо уточнить, что все операции записи (в память, в регистр статуса и в бит разрешения записи) начинают реально выполняться только после подачи признака конца транзакции (CS = «1»), и если он не будет подан, то соответствующая команда не будет выполнена.

Чтение

В случае, когда выполняется чтение (рис. 2), после старта транзакции (CS = «0») посылается 8-битный опкод команды READ; потом следует адрес ячейки, из которой нужно извлечь значение байта (8-битный для

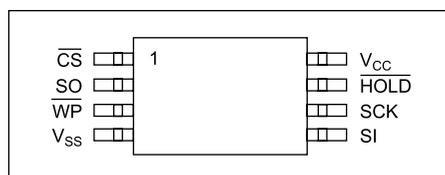


Рис. 1. Расположение выводов CAT25xxx

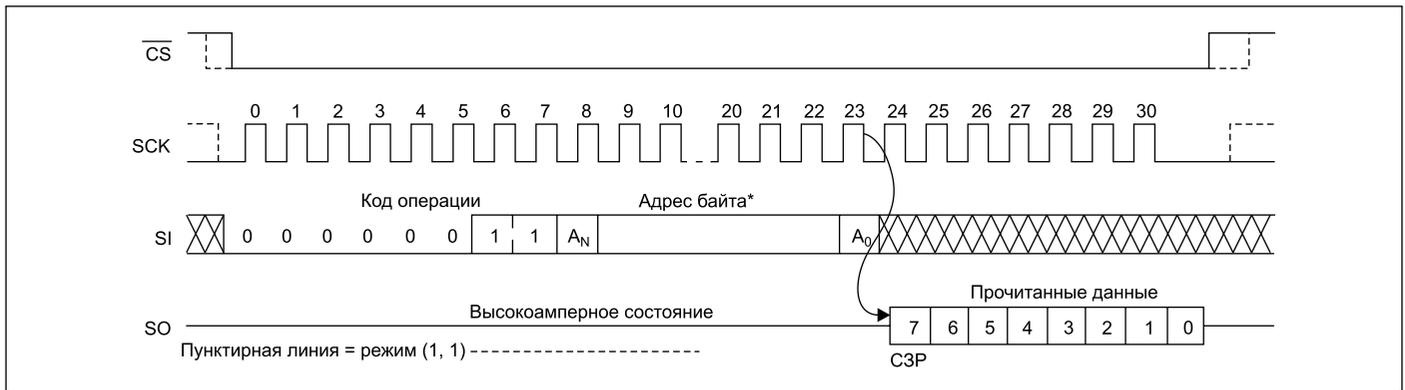


Рис. 2. Временные диаграммы процесса чтения

микросхем с 1, 2, 4 кбит памяти, 16-битный для остальных); при следующем тактовом импульсе на выходе SO появится старший бит запрошенного байта, затем каждый следующий такт будет появляться по одному новому биту. Причем если продолжать подавать тактовый сигнал, то на выходе будут появляться биты из следующих байтов вплоть до конца памяти, при достижении которого указатель адреса считывания будет установлен на нулевую ячейку и считывание пойдет дальше. Таким образом, возможно последовательно считать хоть всю память с помощью одной команды. Прерывание считывания производится с помощью установки CS в состояние «лог. 1».

Запись

Перед любой операцией записи (как в память, так и в регистр статуса) необходимо ввести бит разрешения записи (WEL (Write Enable), при включении и после окончания процесса записи он сброшен) с помощью команды $WREN$ (устанавливаем $CS = \langle 0 \rangle$, подаем опкод $WREN$, $CS = \langle 1 \rangle$). После этого можно приступить собственно к записи (рис. 3), которая производится в следующем порядке: старт транзакции ($CS = \langle 0 \rangle$); опкод команды $WRITE$; адрес ячейки для записи (8-битный для микросхем с 1, 2, 4 кбит памяти, 16-битный для остальных); 8 бит данных для записи; стоп транзакции ($CS = \langle 1 \rangle$). Необходимо учесть, что реально внутренний цикл записи начинается толь-

ко после установки CS в состояние «лог. 1». Соответственно, если этого не сделать, то записи не произойдет.

Микросхема оснащена буфером, позволяющим производить последовательную запись в соседние ячейки памяти без необходимости посылать адрес для каждого байта памяти: отправляется адрес первой ячейки, а дальше внутренний указатель инкрементируется автоматически. Эта возможность называется страничной записью. В этом случае вместо того, чтобы установить конец транзакции, надо продолжать подавать тактовый сигнал и синхронно выдавать новые биты для записи, организованные по 8 бит. Внутренний указатель адреса ячейки для записи будет автоматически увеличиваться на единицу. Это, как уже было сказано, возможно благодаря наличию внутреннего буфера, размер которого и определяет количество байт, которые можно непрерывно записывать следующим образом: 16 байт для устройств с 1, 2, 4 кбит памяти, 32 байта для микросхем с 8 и 16 кбит памяти, 64 байта для остальных. После установки CS в состояние «лог. 1» будет запущен сам процесс записи.

Однако тут есть одна тонкость, ведь не зря функция называется страничной записью. Известно, что внутренняя память является не однородным массивом, а разбита на отдельные части — страницы. Адреса всех байтов, записанных указанным образом, должны находиться в пределах одной физической страницы. В случае, если при записи будет

достигнут конец страницы, возникнет «переполнение» и следующий байт запишется в нулевую ячейку текущей, а не следующей страницы. Тут нужно внимательно следить, чтобы не повредить уже записанные данные и не перепутать, что где было записано.

Пока не завершена запись, все, что можно делать, — считать регистр статуса командой $RDSR$. Это может быть полезно для контроля за временем завершения записи — бит RDY (Ready) будет установлен, пока микросхема занята, и сброшен по окончании процесса.

Защита от записи

Для защиты данных от случайной перезаписи микросхема имеет ряд методов аппаратной и программной защиты.

Во-первых, пока не установлен бит разрешения записи WEL (Write Enable), ни в одну ячейку или регистр запись вообще невозможна.

Во-вторых, можно блок ячеек перевести в «защищенное» состояние, тогда их изменение запрещено даже при установленном бите WEL . Таким образом возможно блокировать 1/4 и 1/2 часть памяти или всю ее целиком. Это действие производится путем установки соответствующего значения битов $BP1$, $BP0$ (табл. 1) в регистре статуса с помощью команды $WRSR$ (Write Status Register).

В-третьих, чтобы предотвратить непреднамеренное изменение регистра статуса, его также можно заблокировать от записи с по-

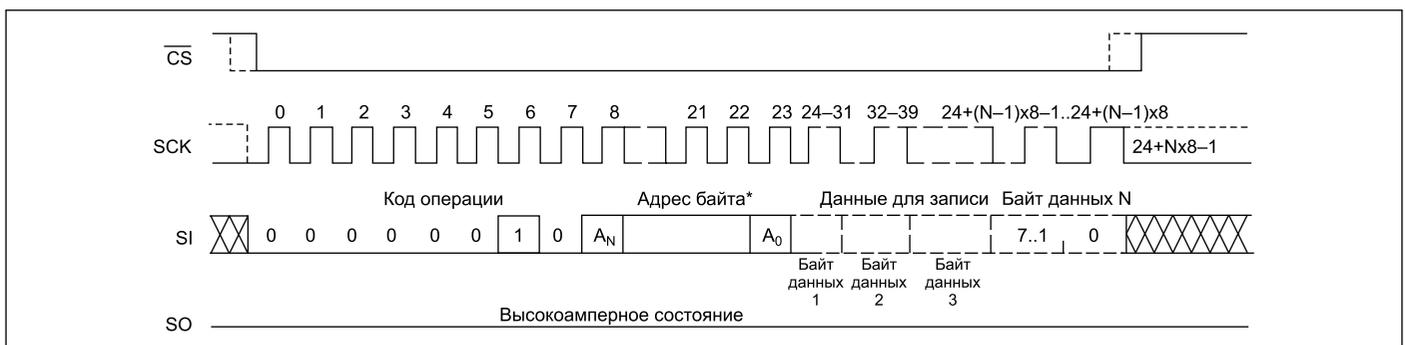


Рис. 3. Временные диаграммы процесса записи

Таблица 1. Защита массива памяти

BP1	BP0	Защищаемые адреса
0	0	Ничего
0	1	1/4 из старших адресов
1	0	1/2 из старших адресов
1	1	Вся память

Таблица 2. Состояние защиты при разных комбинациях параметров

Бит WEL	Бит WPEN	Выход WP	Защищенные блоки	Незащищенные блоки	Регистр статуса
0	x	x	запрещена	запрещена	запрещена
1	0	x	запрещена	разрешена	разрешена
1	1	0	запрещена	разрешена	запрещена
1	1	1	запрещена	разрешена	разрешена

мощью перевода вывода микросхемы WP в низкое состояние. Однако предварительно необходимо установить бит WPEN (Write-Protect Enable) в регистре статуса, который работает как предохранитель для вывода WP: когда WPEN сброшен, состояние WP не имеет значения. Таким образом, защита от записи регистра статуса работает только когда установлен WPEN и WP притянут к «лог. 0». Более наглядно разобраться в возможных состояниях защиты от записи поможет таблица 2 [4].

Серия CAT24Cxxx

Серия CAT24Cxxx (рис. 4) схожа по функциональности с вышеописанными приборами. Основным отличием является то, что данные устройства работают по интерфейсу I²C на частотах 100 кГц (Standart), 400 кГц (Fast) и 1 МГц (Fast-plus), в связи с чем приборы имеют более низкое энергопотребление (1 мА чтение, 3 мА запись) [5]. CAT24Cxxx являются pin-to-pin заменой для микросхем серий AT24C (Atmel), 24LC (Microchip).

Назначение выводов у 8-выводных микросхем серии CAT24Cxxx следующее: два вывода питания V_{CC} (+), V_{SS} (-); пять входов тактового сигнала SCL (Serial Clock); адреса A₂, A₁, A₀ защиты от записи WP (Write Protect); один вывод SDA (Serial Data) для данных, работающий как на вход, так и на выход (учтите, что это вывод с открытым коллектором). Как видно, большинство выводов тут иные, чем в серии CAT25xxx, что связано с использованием другого интерфейса и, соответственно, других принципов коммуникации между ведущим (микроконтроллер) и ведомым (микросхема памяти) устройствами.

Принципы организации ввода/вывода

Так как в протоколе I²C [6] использует один провод и для вывода, и для ввода информации, то устройства должны работать попеременно то на прием, то на передачу. Арбитром, определяющим поведение

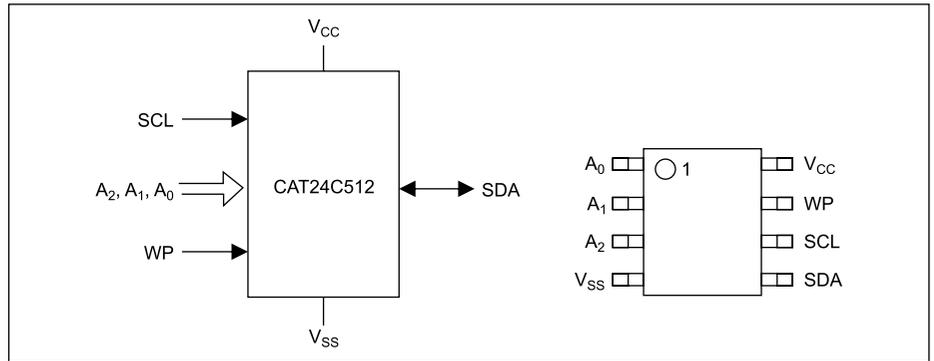


Рис. 4. Расположение выводов CAT24Cxxx

устройство в данный момент времени, является ведущий (Master), который передает для этого специальные команды СТАРТ и СТОП.

Состояние начала передачи СТАРТ определяется с помощью перевода линии данных (SDA) из «лог. 1» в «лог. 0» в то время, когда на тактовой линии (SCL) высокий уровень. Состояние конца передачи СТОП определяется наоборот, при переводе линии данных из «лог. 0» в «лог. 1», когда на SCL опять же высокий уровень. По получении команды СТОП начинается цикл записи (в случае записи) или устройство переводится в режим пониженного энергопотребления (в случае чтения). Поэтому важно помнить, что во всех остальных случаях при приеме и передаче сигнал на линии данных (SDA) должен изменяться только в то время, когда тактовый сигнал находится в низком состоянии («0»), иначе он будет ложно интерпретирован как СТАРТ или СТОП.

К одним и тем же линиям данных SDA и тактирования SCL может быть подключено несколько устройств — за счет того, что каждому из них назначается свой адрес путем подтяжки соответствующих выводов A₀, A₁, A₂ к «лог. 0» или «лог. 1». Далее в каждом сеансе связи Master передает адрес устройства, к которому он обращается для выполнения очередной команды. Адрес состоит из восьми битов: четыре старших равны «1010», три следующих составляют собственно адрес устройства (он должен быть равным состоянию выводов A₂, A₁, A₀), последний — R/W — определяет, будет ли производиться

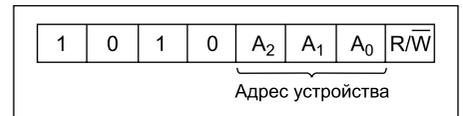


Рис. 5. Адресация ведомого устройства памяти

чтение (1) или запись (0) (рис. 5). Не забудьте, что линии SDA и SDL должны быть подтянуты через резистор к «лог. 1», поскольку соответствующие выводы в микросхеме устроены по схеме с открытым коллектором.

Для уменьшения количества сбоев протокол предусматривает запрос подтверждения (ACK) от приемника, который производится следующим образом: после каждых восьми тактов передачи данных передатчик останавливается, отпускает линию SDA и ожидает, что приемник переведет линию данных в состояние «лог. 0» на время девятого цикла. Если этого не происходит, то сессия прерывается и мастер посылает команду СТОП.

Чтение

Для выполнения процесса чтения используется несколько необычный механизм (рис. 6). Дело в том, что внутренний указатель адреса ячейки один и тот же для чтения и для записи. Поэтому в нем содержится адрес последней ячейки, с которой производили какую-либо из этих операций, плюс 1. При получении команды чтения микросхема выдаст значение байта, на который указывает указатель. Если последняя операция (чтения или записи) производилась над N-й

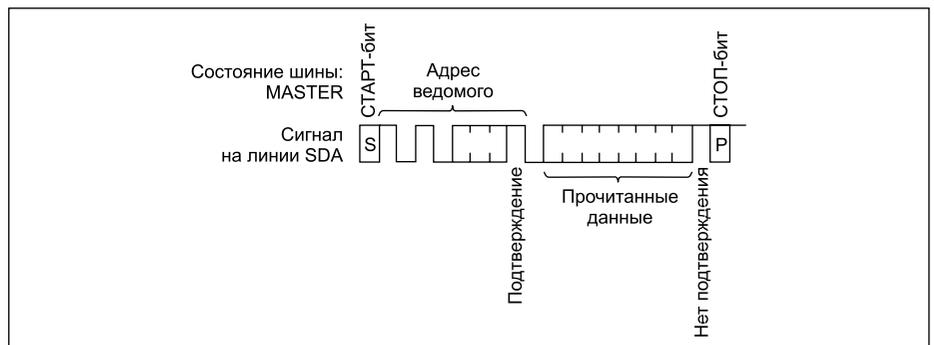


Рис. 6. Временные диаграммы процесса непосредственного чтения

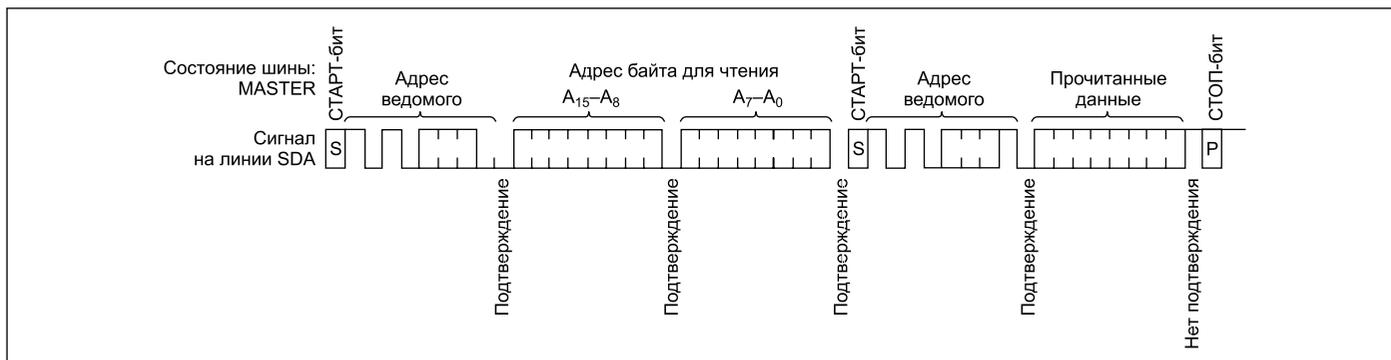


Рис. 7. Временные диаграммы процесса чтения по заданному адресу

ячейкой в массиве памяти, то будет выдано значение, хранящееся в $N+1$ ячейке. Такой вид считывания называется непосредственным чтением (Immediate Address Read). Последовательность при этом такова: подается СТАРТ, потом адрес ведомого устройства (Slave address, последний бит должен быть равен 1 — «чтение»), подтверждение (ACK), на выходе получаем запрошенный байт. Если мы хотим продолжить последовательное считывание памяти, то подаем ACK и получаем следующий байт и т. д. Если хотим завершить, то не подаем ACK, а выжидаем один цикл и подаем СТОП.

Для выполнения выборочного чтения по заданному адресу (Selective Read) необходимо сначала записать этот адрес во внутренний указатель (рис. 7). Это делается с помощью операции «фальшивой» записи, когда последовательность действий такая же, как при записи (бит R/W = «0»), однако передается только адрес ячейки для записи, не передается никаких данных, а немедленно после адреса производится процесс непосредственного чтения (так как адрес уже занесен в указатель).

Запись

Запись производится путем подачи команды СТАРТ, далее следует адрес ведомого устройства (Slave address, последний бит должен быть равен 0 — «запись»), подтверждение (ACK), адрес ячейки для записи (один или два байта), ACK, байт данных, ACK, СТОП (рис. 8). Так же, как в случае чтения, если мы не подаем СТОП после ACK, то можем

последовательно записать несколько байт. Максимальное количество записываемых таким образом данных ограничено одной страницей памяти. Последовательность завершается подачей СТОП, после чего начинается внутренний цикл записи, во время которого микросхема не будет отвечать на любые команды чтения или записи.

Есть один любопытный момент: не важно, записывается последовательно один байт или множество — все равно будут обновлены все ячейки на странице. Измененные получат новые данные, а неизменные будут перезаписаны своими же значениями. Поэтому длительность цикла записи не зависит от количества байт, последовательно записываемых в одной команде.

Защита от записи

Эта функция организована проще, чем в серии CAT25, когда WP находится в состоянии «лог. 1» и вся память защищена от записи. Вывод имеет внутренние резисторы, притягивающие его к потенциалу «лог. 0», поэтому когда он оставлен не подключенным, то защита от записи отключена.

Взаимозаменяемость микросхем и аналоги

В заключение приводим краткую таблицу соответствия моделей микросхем от ON Semiconductor наиболее популярным моделям EEPROM от Microchip и Atmel (таблица 3).

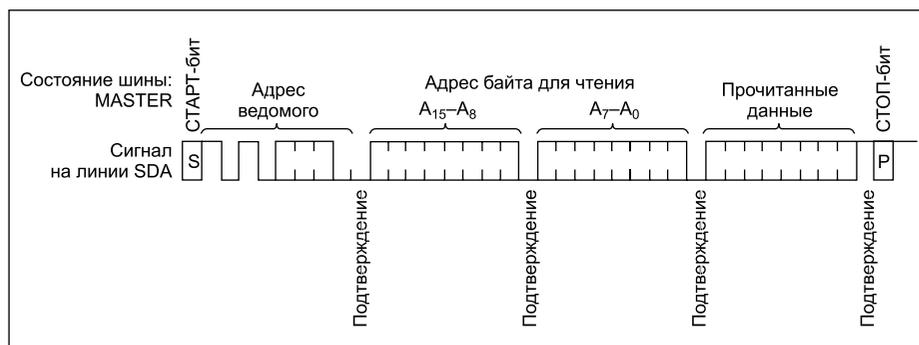


Рис. 8. Тайминги процесса записи

Литература

- <http://www.ixbt.com/news/hard/index.shtml?14/76/20>
- <http://catalog.compel.ru>
- CAT25256. 256-Kb SPI Serial CMOS EEPROM // Semiconductor Components Industries, LLC. 2010. Rev. 4.
- 25AA256/25LC256, 256K SPI Bus Serial EEPROM // Microchip Technology Inc. 2005. Rev. E.
- CAT24C512. 512 kb I²C CMOS Serial EEPROM // Semiconductor Components Industries, LLC. 2011. Rev. 1.
- UM10204. I²C-bus specification and user manual // NXP Semiconductors. 2007. Rev.03.

Таблица 3. Аналоги микросхем памяти EEPROM

Исходная модель	Аналог ON Semiconductor
Atmel	
AT25256B-SSHL-B	CAT25256VI-GT3
AT24C512BN-SH25-B	CAT24C512WI-GT3
AT24C16BN-SH-T	CAT24C16WI-GT3
AT24C08BN-SH-T	CAT24C08WI-GT3
AT24C1024BN-SH25-B	CAT24M01WI-GT3
AT24C04BN-SH-T	CAT24C04WI-GT3
AT24C128BN-SH-T	CAT24C128WI-GT3
AT24C512BN-SH25-T	CAT24C512WI-GT3
AT24C1024BW-SH25-B	CAT24M01WI-GT3
AT24C02C-SSHM-T	CAT24C02WI-G
AT24C256C-SSHL-T	CAT24C256WI-GT3
Microchip	
24LC512T-1/SM	CAT24M01WI-GT3
24LC04BT-1/SN	CAT24C04WI-GT3
24LC512-1/P	CAT24C512LI-G
24LC256-1/P	CAT24C256LI-G
24LC64-1/P	CAT24C64LI-G
24LC64T-1/SN	CAT24C64WI-GT3
24LC16BT-1/SN	CAT24C16WI-GT3
24LC256-1/SM	CAT24C256WI-GT3
24LC515-1/SM	CAT24C512WI-GT3
24LC08B-1/P	CAT24C08LI-G
24LC128T-1/SN	CAT24C128WI-GT3
24LC16B-1/P	CAT24C16LI-G
24LC256-1/SN	CAT24C256WI-GT3
25LC320-1/P	CAT25320LI-G
24LC1025-1/SM	CAT24M01WI-GT3
24LC512-1/SM	CAT24C512WI-GT3
25LC640-1/SN	CAT25640VI-GT3
24LC02B-1/SN	CAT24C02WI-GT3
24LC256T-1/SM	CAT24C256
24LC01BT-1/SN	CAT24C01WI-GT3
24LC256-E/P	CAT24C256LE-G
25LC1024-1/SM	CAT25M01WI-GT3
24LC256T-1/SN	CAT24C256WI-GT3
25LC256-1/SN	CAT25256VE-GT3
25LC040A-1/P	CAT25040LE-G