

# Лабораторная работа N5

## Изучение контроллера Ethernet

### Оглавление

Интерфейс Ethernet, общие сведения .....	2
Основные термины .....	2
Методы кодирования, применяемые в Ethernet .....	3
Метод кодирования NRZI .....	3
Метод кодирования MLT-3 .....	3
Формат кадра IEEE 802.3 .....	4
Доступ к каналу в Ethernet 10 Мбит .....	5
Доступ к каналу в Fast Ethernet .....	7
Физический уровень Fast Ethernet .....	9
Контроллер Ethernet LAN91C111 .....	11
Общие сведения .....	11
Структурная схема .....	11
Регистры физического уровня .....	12
Доступ к регистрам физического уровня .....	12
Регистр управления – адрес 0 .....	14
Регистры MAC уровня .....	16
BANK- Регистр выбора банка .....	16
Банк 0, TCR - регистр управления передачей .....	16
Банк 0, RCR - регистр управления приемом .....	17
Банк 0, RPCR - регистр управления приемом на физическом уровне .....	17
Банк 1, CONFIG - регистр конфигурации .....	18
Банк 2, MMU COMMAND - регистр команд MMU .....	18
Банк 2, регистр номера пакета .....	19
Банк 2, FIFO PORTS - регистр кольцевых буферов .....	19
Банк 2, POINTER - регистр указатель .....	20
Банк 2, DATA - регистр данных .....	20
Банк 2, INTERRUPT - регистр статуса прерываний .....	20
Банк 3, регистр MMI интерфейса .....	21
Инициализация контроллера LAN91C111 .....	22
Сброс контроллера .....	22
Инициализация блока PHY .....	22
Инициализация приема и передачи .....	22
Чтение пакета .....	22
Запись пакета .....	23
Литература .....	24

## Интерфейс Ethernet, общие сведения

Сеть Ethernet разработана в 1976 году Меткальфом и Боггсом (фирма Ксерокс) [2]. Ethernet совместно со своей скоростной версией Fast Ethernet, GigaEthernet (1Гбит/с) и 10GE (10Гигабит/с) занимает в настоящее время абсолютно лидирующую позицию. В настоящее время на основе этого стандарта строятся уже не только локальные но и общегородские сети, а также межгородские каналы. Единственным недостатком данной сети является отсутствие гарантии времени доступа к среде (и механизмов, обеспечивающих приоритетное обслуживание), что делает сеть малоперспективной для решения технологических проблем реального времени. Определенные проблемы иногда создает ограничение на максимальное поле данных, равное ~1500 байт.

### Основные термины

**LLC** – Logical Line Control, управление логической связью. LLC, функции которого определены стандартом IEEE 802.2, фактически обеспечивает взаимосвязь с протоколами более высокого уровня, (например, с IP или IPX), предоставляя различные коммуникационные услуги:

- Сервис без установления соединения и подтверждений приема. Простой сервис, который не обеспечивает управления потоком данных или контроля ошибок, а также не гарантирует правильную доставку данных.
- Сервис с установлением соединения. Абсолютно надежный сервис, который гарантирует правильную доставку данных за счет установления соединения с системой-приемником до начала передачи данных и использования механизмов контроля ошибок и управления потоком данных.
- Сервис без установления соединения с подтверждениями приема. Средний по сложности сервис, который использует сообщения подтверждения приема для обеспечения гарантированной доставки, но не устанавливает соединения до передачи данных.

**MAC** – Media Access Control, управление доступом к среде. Каждый узел в сети Fast Ethernet имеет контроллер доступа к среде (Media AccessController — MAC). MAC имеет ключевое значение в Fast Ethernet и имеет три назначения:

- определяет, когда узел может передать пакет;
- пересылает кадры уровню РНУ для преобразования в пакеты и передачи в среду;\*
- получает кадры из уровня РНУ и передает обрабатывающему их программному обеспечению (протоколам и приложениям).\*

**МII** – Media Independent Interface, интерфейс независимый от среды.

**РНУ** – Physical Layer Device, устройство физического уровня. Поскольку Fast Ethernet может использовать различный тип кабеля, то для каждой среды требуется уникальное предварительное преобразование сигнала. Преобразование также требуется для эффективной передачи данных: сделать передаваемый код устойчивым к помехам, возможным потерям, либо искажениям отдельных его элементов (бодов), для обеспечения эффективной синхронизации тактовых генераторов на передающей или приемной стороне.

**PCS** – Physical Coding Sublayer, подуровень кодирования. Кодирует/декодирует данные поступающие от/к уровня MAC с использованием алгоритмов 4B/5B или 8B/6T.

**РМА** – Physical Medium Attachment, подуровень физического присоединения

**РМД** – Physical Medium Dependent, подуровень зависимости физической среды. Подуровни РМА и РМД осуществляют связь между подуровнем PSC и интерфейсом MDI, обеспечивая формирование в соответствии с методом физического кодирования: NRZI или MLT-3.

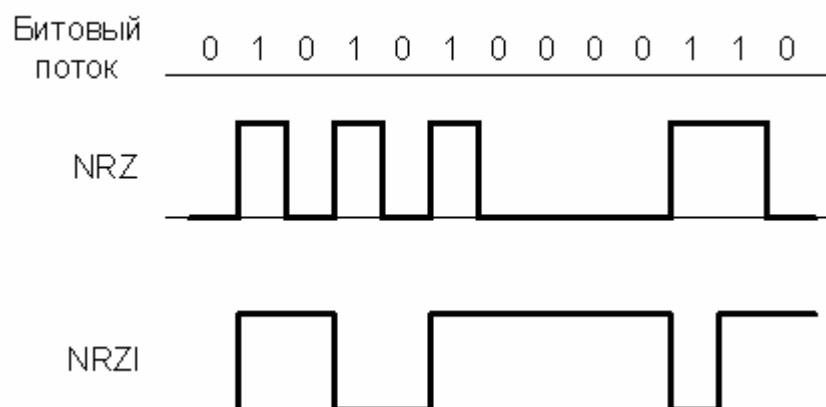
**AUTONEG** – Auto-Negotiation – подуровень переговоров о скорости передачи.

**MDI** – Medium Dependent Interface – разъем сетевого интерфейса. Подуровень автопереговоров позволяет двум взаимодействующим портам автоматически выбирать наиболее эффективный режим работы: дуплексный или полудуплексный 10 или 100 Мб/с.

## **Методы кодирования, применяемые в Ethernet**

### **Метод кодирования NRZI**

NRZI – Non Return to Zero Invertive (инверсное кодирование без возврата к нулю) Этот метод является модифицированным методом Non Return to Zero (NRZ), где для представления 1 и 0 используются потенциалы двух уровней. В коде NRZ I также используется 2 потенциала, но его текущее значение зависит от предыдущего. Если текущее значение бита “1”, то полученный потенциал должен быть инверсией от предыдущего, если значение бита “0” – такой же.

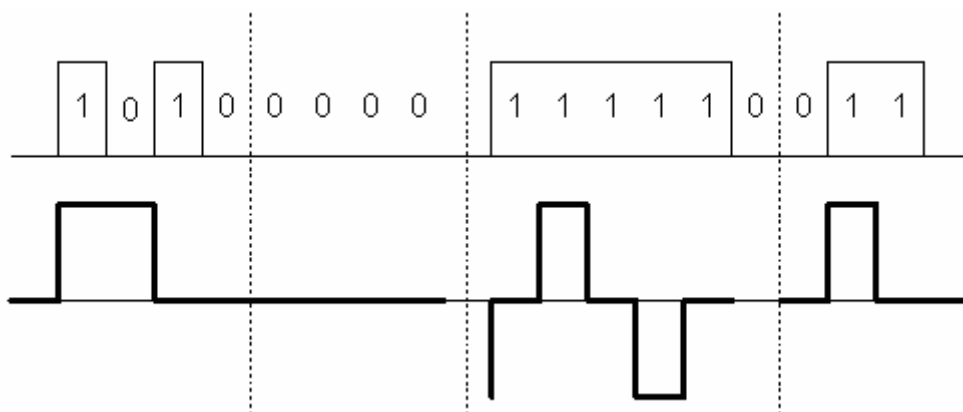


Поскольку код незащищен от долгих последовательностей “нулей” или “единиц”, то это может привести к проблемам синхронизации. Поэтому перед передачей, заданную последовательность битов рекомендуется предварительно закодировать кодом предусматривающим скремблирование (скремблер предназначен для придания свойств случайности передаваемой последовательности данных с целью облегчения выделения тактовой частоты приемником).

### **Метод кодирования MLT-3**

MLT-3 Multi Level Transmission – 3 (многоуровневая передача) - немного схож с кодом NRZ, но в отличие от последнего имеет три уровня сигнала.

Единице соответствует переход с одного уровня сигнала на другой, причем изменение уровня сигнала происходит последовательно с учетом предыдущего перехода. При передаче “нуля” сигнал не меняется.



Этот код, так же как и NRZ нуждается в предварительном кодировании.

### Формат кадра IEEE 802.3

Преамбула	SFD	Адрес получателя	Адрес отправителя	Протокол или тип	Данные до 1500 байт	CRC	EFD
-----------	-----	------------------	-------------------	------------------	---------------------	-----	-----

Рисунок 1 Формат кадра IEEE 802.3

Поле преамбула содержит 7 байт 0xAA и служит для стабилизации и синхронизации среды (чередующиеся сигналы CD1 и CD0 при завершающем CD0), далее следует поле SFD (start frame delimiter = 0xAB), которое предназначено для выявления начала кадра. Поле EFD (end frame delimiter) задает конец кадра. Поле контрольной суммы (CRC - cyclic redundancy check), также как и преамбула, SFD и EFD, формируются и контролируются на аппаратном уровне. В некоторых модификациях протокола поле EFD не используется. Пользователю доступны поля, начиная с адреса получателя и кончая полем информация, включительно. После CRC следует межпакетная пауза (IPG - interpacket gap - межпакетный интервал) длиной 9,6 мксек или более (для 10 мегабитной сети). Максимальный размер кадра равен 1518 байт (сюда не включены поля преамбулы, SFD и EFD). Интерфейс просматривает все пакеты, следующие по кабельному сегменту, к которому он подключен, ведь определить, корректен ли принятый пакет и кому он адресован, можно лишь приняв его целиком. Корректность пакета по CRC, по длине и кратности целому числу байт производится после проверки адреса места назначения. Вероятность ошибки передачи при наличии crc контроля составляет ~2-32. При вычислении CRC используется образующий полином:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1.$$

Алгоритм вычисления CRC сводится к вычислению остатка от деления кода  $M(x)$ , характеризующего кадр, на образующий полином  $G(x)$  (Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specification. Published by IEEE (802.3-1985). Wiley-Interscience, John & sons, inc.). CRC представляет собой дополнение полученного остатка  $R(x)$ . CRC пересылается, начиная со старших разрядов.

Минимальная длина пакета должна быть больше удвоенного значения этой задержки (выбрано 64 байта = 512 тактов). Если размер пакета меньше 64 байт, добавляются байты-

заполнители, чтобы кадр в любом случае имел соответствующий размер. При приеме контролируется длина пакета и, если она превышает 1518 байт, пакет считается избыточным и обрабатываться не будет. Аналогичная судьба ждет кадры короче 64 байт. Любой пакет должен иметь длину, кратную 8 бит (целое число байт). Если в поле адресата содержатся все единицы, адрес считается широковещательным, то есть обращенным ко всем рабочим станциям локальной сети. Пакет ethernet может нести от 46 до 1500 байт данных. Формат адреса получателя или отправителя (MAC) показан на Рисунок 2. Для передачи данных на физическом уровне используется манчестерский код.

47	0
1	1
22	24
I/G	U/L
OUI	Адрес (OUA)

**Рисунок 2 Формат mac-адреса**

В верхней части рисунка указана длина полей адреса, в нижней - нумерация разрядов. Субполе I/G представляет собой флаг индивидуального или группового адреса. I/G=0 - указывает на то, что адрес является индивидуальным адресом сетевого объекта. I/G=1 характеризует адрес как мультикастинговый, в этом случае дальнейшее разбиение адреса на субполя теряет смысл. Субполе UL является флагом универсального или местного управления (определяет механизм присвоения адреса сетевому интерфейсу). U/L=1 указывает на локальную адресацию (адрес задан не производителем и ответственность за уникальность лежит на администраторе LAN). U/L=I/G=0 характерно для стандартных уникальных адресов, присваиваемых интерфейсу его изготовителем. Субполе OUI (organizationally unique identifier) позволяет определить производителя сетевого интерфейса. Каждому производителю присваивается один или несколько OUI. Размер субполя позволяет идентифицировать около 4 миллионов различных производителей. За корректность присвоения уникального адреса интерфейса (OUA - organizationally unique address) несет ответственность производитель. Двух интерфейсов одного и того же производителя с идентичными номерами не должно существовать. Размер поля позволяет произвести примерно 16 миллионов интерфейсов. Комбинация oui и oua составляют UAA (universally administrated address = IEEE-адрес).

Если в поле кадра протокол/тип записан код менее 1500, то это поле характеризует длину кадра. В противном случае - это код протокола, пакет которого инкапсулирован в кадр Ethernet.

## **Доступ к каналу в Ethernet 10 Мбит**

Доступ к каналу Ethernet базируется на алгоритме CSMA/CD (carrier sense multiple access with collision detection). В Ethernet любая станция, подключенная к сети, может попытаться начать передачу пакета (кадра), если кабельный сегмент, к которому она подключена, свободен. Свободен ли сегмент, интерфейс определяет по отсутствию "несущей" в течение 9,6 мксек. Так как первый бит пакета достигает остальных станций сети не одновременно, может случиться, что попытку передачи совершат две или более станций, тем более что задержки в повторителях и кабелях могут достигать достаточно больших величин. Такие совпадения попыток называются столкновениями. Столкновение (коллизия) распознается по наличию в канале сигнала, уровень которого соответствует работе двух или более трансиверов одновременно. При обнаружении столкновения станция прерывает передачу. Возобновление попытки может быть произведено после выдержки (кратной 51,2 мксек, но не превосходящей 52 мсек), значения которой является псевдослучайной величиной и вычисляется каждой станцией независимо ( $t = \text{RAND}(0, 2\text{min}(n, 10))$ ), где  $n$  - содержимое счетчика попыток, а число 10 - backofflimit).

После выдержки станция увеличивает на единицу счетчик попыток и начинает очередную передачу. Предельное число попыток по умолчанию равно 16, если число попыток исчерпано, связь прерывается и выдается соответствующее сообщение. Передаваемый

длинный кадр способствует "синхронизации" начала передачи пакетов несколькими станциями. Ведь за время передачи с заметной вероятностью может возникнуть необходимость передачи у двух и более станций. В момент, когда они обнаружат завершение пакета, будут включены таймеры IPG. К счастью информация о завершении передачи пакета доходит до станций сегмента не одновременно. Но задержки, с которыми это связано, являются также причиной того, что факт начала передачи нового пакета одной из станций не становится известным немедленно. При вовлечении в столкновение нескольких станций они могут уведомить остальные станции об этом, пошлав сигнал "затора" (jam - не менее 32 бит). Содержимое этих 32 бит не регламентируется. Такая схема делает менее вероятным повторное столкновение. Источником большого числа столкновений (помимо информационной перегрузки) может служить запредельная суммарная длина логического кабельного сегмента, слишком большое число повторителей, обрыв кабеля или неисправность одного из интерфейсов. Но сами по себе столкновения не являются чем-то негативным - это механизм, регулирующий доступ к сетевой среде.

Под логическим кабельным сегментом (иногда называемым областью столкновений) подразумевается один или несколько кабельных сегментов, объединенных повторителями. Анализ столкновений является одним из средств эффективной диагностики сети. Локальные столкновения (столкновения на сегменте, к которому непосредственно подключена рабочая станция) порождают укороченные пакеты-фрагменты (ведь их передача прерывается) с длиной менее 64 октетов. Большинство трансиверов и репитеров имеют на своих передних панелях индикаторы столкновений. Диаграмма состояний CSMA/CD показана на Рисунок 3. Большую часть времени канальный уровень находится в состоянии прослушивания линии. При появлении заголовка кадра с нужным адресом получателя происходит переход в состояние приема кадра. Когда кадр принят, канальный уровень сообщает об этом событии более высоким уровням сети и возвращается в состояние прослушивания. Кадр можно передать по запросу сетевого уровня. Когда такой запрос получен, происходит переход в состояние ожидания свободного канала. Если канал свободен, происходит переход в состояние передачи. После возникновения коллизии происходит переход к задержке (разной для разных узлов сети) и, затем, переход в состояние ожидания.

Особое внимание я бы хотел обратить на влияние сигнала jam. В процессе пересылки столкнувшихся пакетов и за время передачи сигнала jam другие узлы могли захотеть что-то передать. Если таких узлов больше одного, то это приведет к синхронизации начала передачи этими узлами и к увеличению вероятности столкновения. Практически такую "синхронизацию" может осуществить любой достаточно длинный пакет. Такая синхронизация является причиной "коллапса" сети при большой загрузке.



Рисунок 3 Диаграмма состояний CSMA/CD

Метод CSMA/CD создает неопределенность времени доступа к сети, что делает ее неудобной для решения некоторых задач управления в реальном масштабе времени, где требуется малое время реакции системы на внешнее воздействие.

### Доступ к каналу в Fast Ethernet

Контроллер MAC Fast Ethernet, прежде чем приступить к передаче, прослушивает несущую. Несущая существует лишь тогда, когда другой узел ведет передачу. Уровень РНУ определяет наличие несущей и генерирует сообщение для MAC. Наличие несущей говорит о том, что среда занята и слушающий узел (или узлы) должны уступить передающему.

MAC, имеющий кадр для передачи, прежде чем передать его, должен подождать некоторый минимальный промежуток времени после окончания предыдущего кадра. Это время называется **межпакетной щелью** (IPG, interpacket gap) и продолжается 0,96 микросекунды, то есть десятую часть от времени передачи пакета обычной Ethernet со скоростью 10 Мбит/с (IPG — единственный интервал времени, всегда определяемый в микросекундах, а не во времени бита) Рисунок 4.

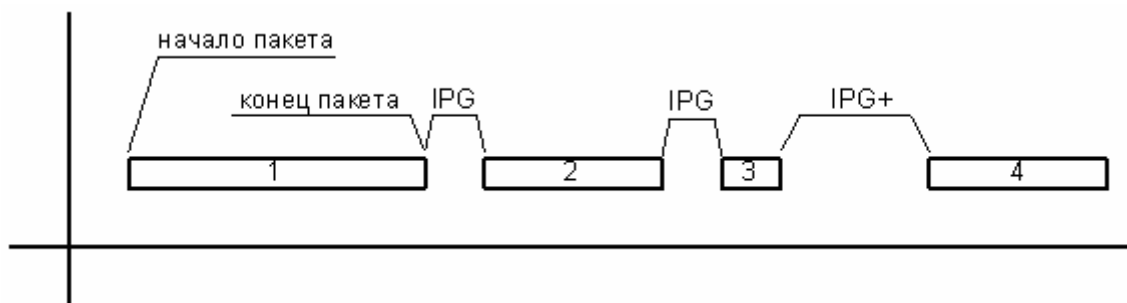


Рисунок 4 Межпакетная щель

После окончания пакета 1 все узлы ЛВС обязаны подождать в течение времени IPG, прежде чем смогут передавать. Временной интервал между пакетами 1 и 2, 2 и 3 на рис. 2 — это время IPG. После завершения передачи пакета 3 ни один узел не имел материала для обработки, поэтому временной интервал между пакетами 3 и 4 длиннее, чем IPG.

Все узлы сети должны соблюдать эти правила. Даже если на узле имеется много кадров для передачи и данный узел является единственным передающим, то после пересылки каждого пакета он должен выждать в течение, по крайней мере, времени IPG.

Именно в этом заключается часть CSMA правил доступа к среде Fast Ethernet. Короче говоря, многие узлы имеют доступ к среде и используют несущую для контроля ее занятости.

В ранних экспериментальных сетях применялись именно эти правила, и такие сети работали очень хорошо. Тем не менее, использование лишь CSMA привело к возникновению проблемы. Часто два узла, имея пакет для передачи и прождав время IPG, начинали передавать одновременно, что приводило к искажению данных с обеих сторон. Такая ситуация называется коллизией (collision) или конфликтом.

Для преодоления этого препятствия ранние протоколы использовали достаточно простой механизм. Пакеты делились на две категории: команды и реакции. Каждая команда, переданная узлом, требовала реакции. Если в течение некоторого времени (называемого периодом тайм-аута) после передачи команды реакция на нее не была получена, то исходная команда подавалась вновь. Это могло происходить по несколько раз (предельное количество тайм-аутов), прежде чем передающий узел фиксировал ошибку.

Эта схема могла прекрасно работать, но лишь до определенного момента. Возникновение конфликтов приводило к резкому снижению производительности (измеряемой обычно в байтах в секунду), потому что узлы часто простаивали в ожидании ответов на команды, никогда не достигающие пункта назначения. Перегрузка сети, увеличение количества узлов напрямую связаны с ростом числа конфликтов и, следовательно, со снижением производительности сети.

Проектировщики ранних сетей быстро нашли решение этой проблемы: каждый узел должен устанавливать факт потери переданного пакета путем обнаружения конфликта (а не ожидать реакции, которая никогда не последует). Это означает, что потерянные в связи с конфликтом пакеты должны быть немедленно переданы вновь до окончания времени тайм-аута. Если узел передал последний бит пакета без возникновения конфликта, значит, пакет передан успешно.

Метод контроля несущей хорошо сочетать с функцией обнаружения коллизий. Коллизии все еще продолжают происходить, но на производительности сети это не отражается, так как узлы быстро избавляются от них. Группа DIX, разработав правила доступа к среде CSMA/CD для Ethernet, оформила их в виде простого алгоритма – Рисунок 5.



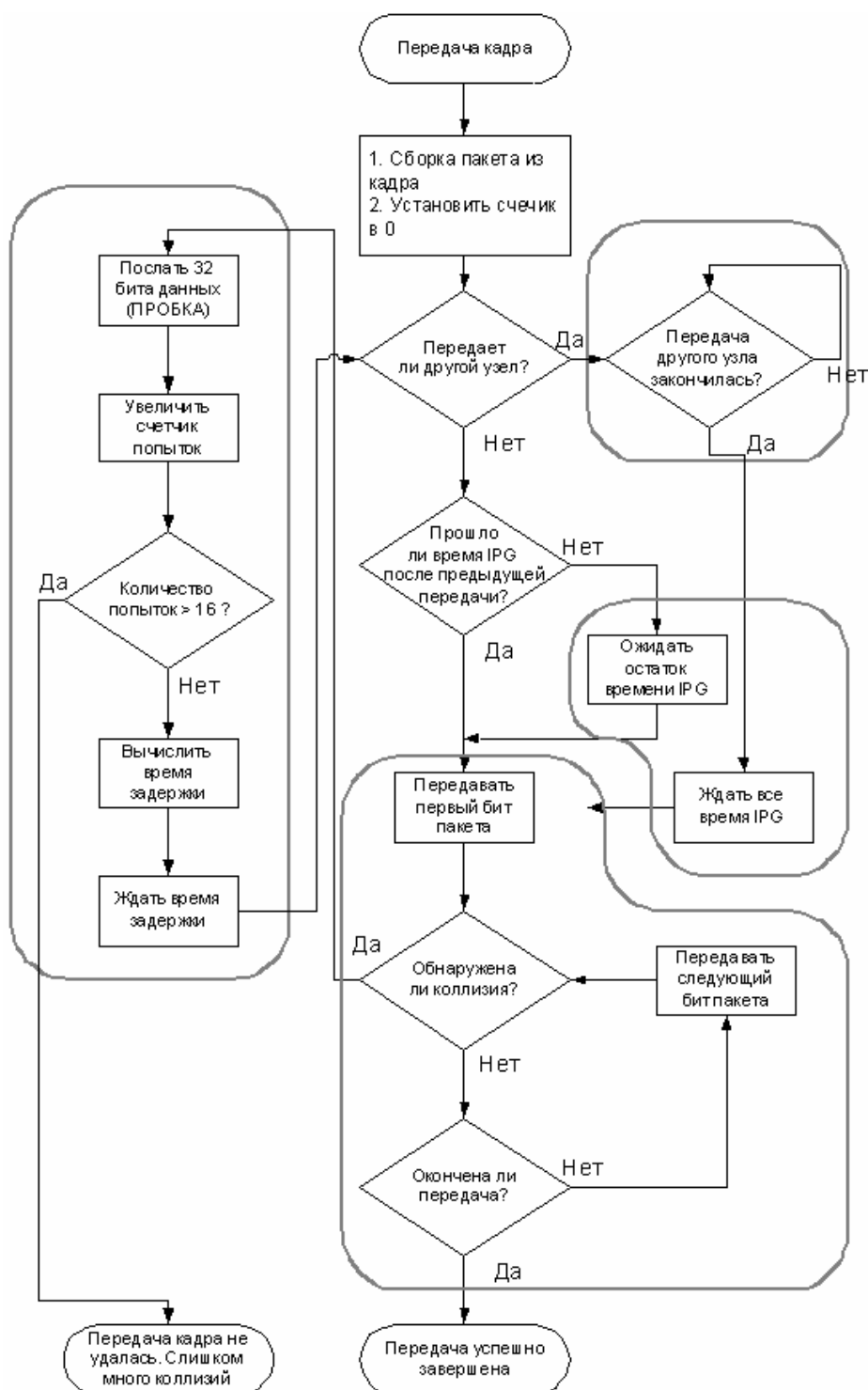


Рисунок 5 Алгоритм работы CSMA/CD

### Физический уровень *Fast Ethernet*

Стандарт Fast Ethernet определяет три типа среды передачи сигналов Ethernet со скоростью 100 Мбит/с.

- 100Base-TX — две витые пары проводов. Передача осуществляется в соответствии со стандартом передачи данных в витой физической среде, разработанным ANSI (American National Standards Institute — Американский национальный институт стандартов). Витой кабель для передачи данных может быть экранированным, либо неэкранированным. Использует алгоритм кодирования данных 4B/5B и метод физического кодирования MLT-3.

- 100Base-FX — две жилы, волоконно-оптического кабеля. Передача также осуществляется в соответствии со стандартом передачи данных в волоконно-оптической среде, которой разработан ANSI. Использует алгоритм кодирования данных 4B/5B и метод физического кодирования NRZI.

Спецификации 100Base-TX и 100Base-FX известны также как 100Base-X

- 100Base-T4 — это особая спецификация, разработанная комитетом IEEE 802.3u . Согласно этой спецификации, передача данных осуществляется по четырем витым парам телефонного кабеля, который называют кабелем UTP категории 3. Использует алгоритм кодирования данных 8B/6T и метод физического кодирования NRZI.

Дополнительно стандарт Fast Ethernet включает рекомендации по использованию кабеля экранированной витой пары категории 1, который является стандартным кабелем, традиционно использующимся в сетях Token Ring. Организация поддержки и рекомендации по использованию кабеля STP в сети Fast Ethernet предоставляют способ перехода на Fast Ethernet для покупателей, имеющих кабельную разводку STP.

Спецификация Fast Ethernet включает также механизм автосогласования, позволяющий порту узла автоматически настраиваться на скорость передачи данных — 10 или 100 Мбит/с. Этот механизм основан на обмене рядом пакетов с портом концентратора или переключателя.

## Контроллер Ethernet LAN91C111

### Общие сведения

Основные характеристики LAN91C111:

- Две скорости обмена – 10/100 Мбит/сек.
- Совместимость с IEEE 802.3/802.3u - 10BASE-TX/10BASE-T
- Автоопределение скорости 10/100 и типа связи: полный дуплекс/полудуплекс.
- Не требует внешних фильтров.
- Поддержка работы в режиме полного дуплекса.
- 8 Кбайт встроенной памяти для организации входной и выходной очереди пакетов.
- Расширенные возможности по управлению энергопотреблением.
- Возможность подключения внешнего конфигурационного EEPROM с последовательным интерфейсом.
- 8, 16 и 32 разрядный режим работы с системной шиной.
- Внутренний 32 разрядный интерфейс к буферу пакетов.

### Структурная схема

Внутри, контроллер LAN91C111 состоит из двух блоков. Блок Ethernet MAC отвечает за прием и формирование пакетов Ethernet. Блок PHY отвечает за формирование сигналов на линиях физического уровня интерфейса Ethernet. Блок PHY соединяется с блоком Ethernet MAC через интерфейс IEEE 802.3 MII (Media Independent Interface).

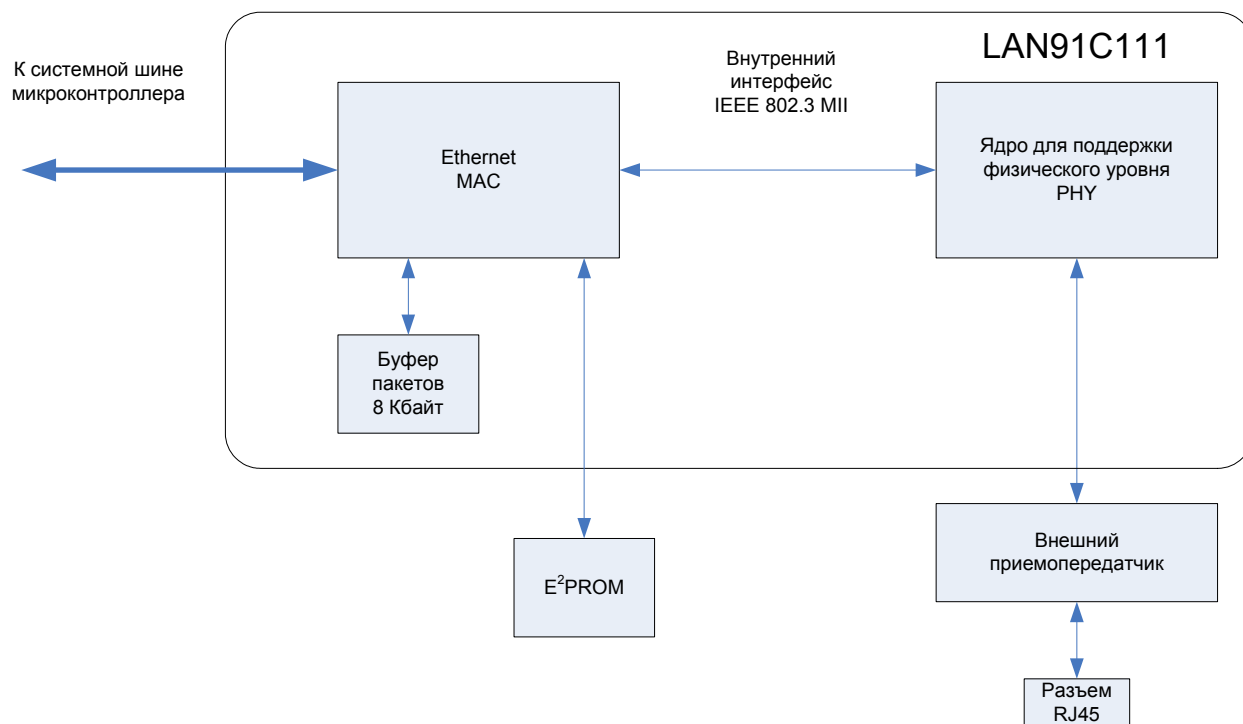


Рисунок 6 Структура контроллера Ethernet LAN91C111

Для увеличения производительности контроллера, к блоку MAC подключен буфер пакетов размеров 8 Кбайт. Увеличение производительности достигается за счет того, что внутренняя шина данных 32 разрядная, а внешний интерфейс для связи с микроконтроллером может быть как 32 разрядным, так и 8 или 16 разрядным. Опционально, к контроллеру Ethernet может подключаться энергонезависимая память для

хранения различных настроек, таких например как MAC адрес контроллера. К блоку РНУ подключается внешний приемопередатчик для обеспечения подключения к шине Ethernet.

## **Регистры физического уровня**

### **Доступ к регистрам физического уровня**

Доступ к блоку РНУ осуществляется посредством интерфейса IEEE 802.3 МП. Этот интерфейс относится к классу последовательных синхронных интерфейсов.

Процедуры доступа к блоку РНУ осуществляются через регистр Management Interface.

```
// Чтение бита из MII
static int mii_read( void )
{
    int res;

    ETH_BANK = 3;

    ETH_MGMT = 0x3330;
    ETH_MGMT = 0x3334;

    res      = ETH_MGMT;

    ETH_MGMT = 0x3330;

    if( res & 0x0002 )
        return 1;
    else
        return 0;
}

// Запись бита в MII или перевод MII на вход (Z)
static void mii_write( unsigned short value )
{
    ETH_BASE = 3;

    switch( value )
    {
        {
        case 0: // '0'
            ETH_MGMT = 0x3338;
            ETH_MGMT = 0x333C;
            ETH_MGMT = 0x3338; break;

        case 1: // '1'
            ETH_MGMT = 0x3339;
            ETH_MGMT = 0x333D;
            ETH_MGMT = 0x3339; break;

        case 2: // 'Z'
            ETH_MGMT = 0x3330;
            ETH_MGMT = 0x3334;
            ETH_MGMT = 0x3330; break;

        }
    }
}

// Начало посылки через MII
static void mii_start( unsigned char mode, unsigned char addr )
{
    int i;

    ETH_BANK = 3;

    // Синхропоследовательность (преамбула) 32 * '1'
    for( i = 0; i < 32; i++ )
    {
        ETH_MGMT = 0x3339;
        ETH_MGMT = 0x333D;
    }

    // Старт-бит '01'

    mii_write( 0 );
}
```

```
mii_write( 1 );

// Операция (read/write)

if( mode == MII_WRITE )
{
    mii_write( 0 );
    mii_write( 1 );
}
else
{
    mii_write( 1 );
    mii_write( 0 );
}

// Адрес устройства PHY (для 91C111 = 0)

for( i = 0; i < 5; i++ )
    mii_write( 0 );

// Адрес регистра

for( i = 0; i < 5; i++ )
{
    if( addr & 0x10 )
        mii_write( 1 );
    else
        mii_write( 0 );

    addr <<= 1;
}

// Переключение направления интерфейса

if( mode == MII_WRITE )
{
    mii_write( 1 );
    mii_write( 0 );
}
else
{
    mii_write( 2 );
}
}
```

Функции доступа к регистрам блока PHY построены на базе представленных выше функций.

```
/**-----
                                read_phy_reg
-----**/

Чтение регистра PHY

Вход:         addr - адрес регистра PHY
Выход:        нет
Результат:   прочитанные данные
Описание:
Пример:
-----*/

unsigned short read_phy_reg( unsigned char addr )
{
    int i;
    unsigned short data;

    mii_start( MII_READ, addr );

    ETH_BANK = 3;

    data = 0;

    for( i = 0; i < 16; i++ )
    {
        data <<= 1;

        if( mii_read() )
```

```

        data |= 1;
    }

    mii_write( 2 );

    return data;
}

/**-----
                                write_phy_reg
-----**/

Запись в регистр PHY

Вход:      addr - адрес регистра PHY
          data - записываемые данные
Выход:     нет
Результат: нет
Описание:
Пример:
-----*/

void write_phy_reg( unsigned char addr, unsigned short data )
{
    int i;

    mii_start( MII_WRITE, addr );

    ETH_BASE = 3;

    for( i = 0; i < 16; i++ )
    {
        if( data & 0x8000 )
            mii_write( 1 );
        else
            mii_write( 0 );

        data <<= 1;
    }

    ETH_MGMT = 0x3330;
}

```

## Регистр управления – адрес 0

RST	LPBK	SPEED	ANEG_EN	PDN	MII_DIS	ANEG_RST	DPLX
RW, SC	RW	RW	RW	RW	RW	RW, SC	RW
0	0	1	1	0	1	0	0

COLST	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0

**RST** – запись «1» в этот бит приводит к сбросу блока PHY. Регистр автоматически сбрасывается в «0» после окончания процедуры сброса. Любая запись в регистр во время сброса будет игнорирована. Время операции сброса занимает 50 мс и на это время необходимо задержать все операции с контроллером.

**LPBK** – бит LOOPBACK для переключения блока PHY в режим проверки. После записи «1» в этот регистр вход замыкается на выход.

**SPEED** – выбор скорости. Если режим автоопределения выключен, то запись «1» приводит к переключению на скорость 100 Мбит/сек, а «0» на скорость 10 Мбит/сек.

**ANEG\_EN** – автоопределение. Если записать «1» в этот бит, то контроллер будет сам определять скорость и тип обмена, а состояние битов **SPEED** и **DPLX** будут игнорироваться.

**PDN** – power down. Запись «1» приводит к выключению блока PHY для экономии энергии. В этом режиме доступен только интерфейс МП.

**MP\_DIS** – выключение интерфейса МП. После сигнала RESET интерфейс МП заблокирован и все команды МП игнорируются блоком PHY. Для обеспечения доступа к блоку PHY необходимо записать в этот бит «0».

**ANEG\_RST** – из этого бита читается «0», если блок PHY в данный момент не поддерживает режим автоопределения. Если этот бит равен «1», то можно сбросить процессы автоопределения в исходное состояние путем записи «1» в этот бит.

**DPLX** – запись «1» в этот бит приводит к переключению в режим полного дуплекса.

**COLST** – запись «1» в этот бит приводит к запуску теста коллизий.

## Регистры MAC уровня

	BANK0	BANK1	BANK2	BANK3
0	TCR	CONFIG	MMU COMMAND	MT0-1
2	EPH STATUS	BASE	PNR	MT2-3
4	RCR	IA0-1	FIFO PORTS	MT4-5
6	COUNTER	IA2-3	POINTER	MT6-7
8	MIR	IA4-5	DATA	MGMT
A	RPCR	GENERAL	DATA	REVISION
C	RESERVED	CONTROL	INTERRUPT	ERCV
E	BANK	BANK	BANK	BANK

Рисунок 7 Регистры MAC уровня

### BANK- Регистр выбора банка

Смещение 0xE.

HIGH BYTE	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	0	0	1	1	0	0	1	1
LOW BYTE						BS2	BS1	BS0
	X	X	X	X	X	0	0	0

BS2..BS0 – номер банка регистров.

### Банк 0, TCR - регистр управления передачей

Смещение 0.

HIGH BYTE	SWFDUP	Reserved	EPH LOOP	STP SQET	FDUPLX	MON_CS	Reserved	NOCRC
	0	0	0	0	0	0	0	0
LOW BYTE	PAD_EN	Reserved	Reserved	Reserved	Reserved	FORCOL	LOOP	TXENA
	0	0	0	0	0	0	0	0

**SWFDUP** – включение работы с коммутатором в режиме полного дуплекса. В этом режиме запрещен анализ коллизий т.к. при работе с сетевым коммутатором в полном дуплексе коллизий не может быть. При установке этого бита FDUPLX и MON\_CS не используются.

**EPH LOOP** – тестовый режим. Отправленные данные возвращаются на собственный приемник.

**STP SQET** – завершение передачи при наличии ошибки SQET.

**FDUPLX** – выключение проверки заголовка принимаемого пакета. В режиме полудуплекса этот бит должен быть установлен «0», т.к. иначе мы будем принимать пакеты отправленные нашим контроллером.

**MON\_CS** – когда этот бит установлен, контроллер проверяет наличие несущей во время передачи.



**NOCRC** – не добавлять CRC к пакету при передаче. Позволяет программному обеспечению самому добавлять CRC к передаваемому пакету.

**PAD\_EN** – если установить этот бит, то контроллер будет автоматически удлинять короткие пакеты до длины 64 байта.

**FORCOL** – (Force a Collision) искусственно вызывает коллизию, при TXENA = 1. Бит управляется программно.

**LOOP** – тестовый режим.

**TXENA** – разрешение передачи.

### Банк 0, RCR - регистр управления приемом

Смещение 4

HIGH BYTE	SOFT RST	FILT CAR	ABORT_ENB	Reserved	Reserved	Reserved	STRIP CRC	RXEN
	0	0	0	0	0	0	0	0
LOW BYTE	Reserved	Reserved	Reserved	Reserved	Reserved	ALMUL	PRMS	RX_ABORT
	0	0	0	0	0	0	0	0

**SOFT RST** – программно-управляемый сброс. Начинается после записи «1» и заканчивается после записи «0».

**FILT CAR** – включение фильтра несущей.

**ABORT\_ENB** – разрешение прерывания приема при наличии коллизии.

**STRIP CRC** – удаляет CRC из принятых пакетов.

**RXEN** – включение приема пакетов.

**ALMUL** – разрешает прием широковещательных (multicast) пакетов.

**PRMS** – включение приема всех пакетов (Promiscuous mode).

**RX\_ABORT** – этот бит устанавливается, если принятый пакет длиннее 2 килобайт. Бит равен 0 после сброса. Сбрасывается программно.

### Банк 0, RPCR - регистр управления приемом на физическом уровне

Смещение 0xA

HIGH BYTE	Reserved	Reserved	SPEED	DPLX	ANEG	Reserved	Reserved	Reserved
	0	0	0	0	0	0	0	0
LOW BYTE	LS2A	LS1A	LS0A	LS2B	LS1B	LS0B	Reserved	Reserved
	0	0	0	0	0	0	0	0

**SPEED** – выбор скорости обмена. Этот бит имеет смысл, если ANEG = 0.

**DPLX** – включение полного дуплекса. Этот бит имеет смысл, если ANEG = 0.

**ANEG** – автоопределение скорости обмена и типа дуплекса.

**LSxx** – биты выбора источников сигнала для светодиодов LEDA и LEDB.

LS2A	LS1A	LS0A	LED SELECT SIGNAL – LEDA
0	0	0	nPLED3+ nPLED0 – Logical OR of 100Mbps Link detected 10Mbps Link detected (default)
0	0	1	Reserved
0	1	0	nPLED0 - 10Mbps Link detected
0	1	1	nPLED1 - Full Duplex Mode enabled
1	0	0	nPLED2 - Transmit or Receive packet occurred
1	0	1	nPLED3 - 100Mbps Link detected
1	1	0	nPLED4 - Receive packet occurred
1	1	1	nPLED5 - Transmit packet occurred

LS2B	LS1B	LS0B	LED SELECT SIGNAL – LEDB
0	0	0	nPLED3+ nPLED0 – Logical OR of 100Mbps Link detected 10Mbps Link detected (default)
0	0	1	Reserved
0	1	0	nPLED0 - 10Mbps Link detected
0	1	1	nPLED1 – Full Duplex Mode enabled
1	0	0	nPLED2 – Transmit or Receive packet occurred
1	0	1	nPLED3 - 100Mbps Link detected
1	1	0	nPLED4 - Receive packet occurred
1	1	1	nPLED5 - Transmit packet occurred

## Банк 1, CONFIG - регистр конфигурации

Смещение – 0.

HIGH BYTE	EPH Power EN	Reserved	Reserved	NO WAIT	Reserved	GPCNTRL	EXT PHY	Reserved
	1	0	1	0	0	0	0	0
LOW BYTE	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	1	0	1	1	0	0	0	1

**EPH Power EN** – управление энергопотреблением EPH. Если значение бита = «0», то EPH переходит в режим пониженного энергопотребления.

**NO WAIT** - если установлена «1» не требуются дополнительные такты неготовности.

**GPCNTR** – бит управляет дискретным портом вывода общего назначения.

**EXT PHY** – бит переключает контроллер на использование внешнего блока PHY.

## Банк 2, MMU COMMAND - регистр команд MMU

Смещение 0

HIGH BYTE							
LOW BYTE	COMMAND		Reserved	Reserved	Reserved	Reserved	BUSY
	Operation Code						
	0						

COMMAND – код команды.

Код	Описание
0	NOOP – нет операции
1	Выделить память в очереди для передачи
2	Сбросить MMU в исходное состояние
3	Убрать пакет с вершины приемной очереди
4	Аналогично команде 3.
5	Освобождает страницы памяти, выделенные для пакета, номер которого указан в регистре PACKET NUMBER REGISTER.
6	Выбрать и передать пакет с номером указанным в регистре PACKET NUMBER REGISTER.
7	Сброс очереди на передачу.

BUSY – равен «1» пока выполняется команда.

## Банк 2, регистр номера пакета

Смещение 2

Reserved	Reserved	PACKET NUMBER AT TX AREA					
0	0	0	0	0	0	0	0

FAILED	Reserved	ALLOCATED PACKET NUMBER					
1	0	0	0	0	0	0	0

FAILED – бит равен «0», если выделение памяти прошло успешно.

## Банк 2, FIFO PORTS - регистр кольцевых буферов

Смещение 4

HIGH BYTE	EMPTY	Reserved	RX FIFO PACKET NUMBER					
	1	0	0	0	0	0	0	0
LOW BYTE	EMPTY	Reserved	TX FIFO PACKET NUMBER					
	1	0	0	0	0	0	0	0

EMPTY – приемной очереди нет принятых пакетов.

RX FIFO PACKET NUMBER – номер пакета находящегося в конце приемного буфера. Значение имеет смысл если бит EMPTY = 0.

EMPTY – очередь пакетов на передачу пуста.

**TX FIFO PACKET NUMBER** – номер пакета находящегося в конце буфера передачи. Значение имеет смысл если бит **EMPTY** = 0.

## Банк 2, POINTER - регистр указатель

Смещение 6

Регистр предназначен для установки указателя в памяти очередей для приема и передачи.

HIGH BYTE	RCV	AUTO INCR.	READ	ETEN	NOT EMPTY	POINTER HIGH		
	0	0	0	0	0	0	0	0
LOW BYTE	POINTER LOW							
	0	0	0	0	0	0	0	0

**RCV** – «1» обращение к приемной очереди, «0» - к передающей.

**AUTOINCR** – включение автоматического инкремента адреса.

**READ** – определяет тип операции: «1» - чтение, «0» - запись.

**ETEN** – включение определения преждевременной передачи пакета (когда пакет еще не до конца сформирован в памяти контроллера).

**NOT EMPTY** – бит равен «1», если буфер передачи не пуст.

## Банк 2, DATA - регистр данных

Смещение 8

DATA HIGH							
X	X	X	X	X	X	X	X
DATA LOW							
X	X	X	X	X	X	X	X

## Банк 2, INTERRUPT - регистр статуса прерываний

Смещение 0xC, только чтение

MDINT	ERCV INT	EPH INT	RX_OVRN INT	ALLOC INT	TX EMPTY INT	TX INT	RCV INT
0	0	0	0	0	1	0	0

Смещение 0xC, только запись

MDINT	ERCV INT		RX_OVRN INT		TX EMPTY INT	TX INT	

Смещение 0xD, чтение и запись

MDINT MASK	ERCV INT MASK	EPH INT MASK	RX_OVRN INT MASK	ALLOC INT MASK	TX EMPTY INT MASK	TX INT MASK	RCV INT MASK
0	0	0	0	0	0	0	0

**MDINT** – устанавливается в «1», если один из битов PHY меняет свое значение: LNKFAIL, LOSSSYNC, CWRD, SSD, ESD, PROL, JAB, SPDDET, DPLXDET

**ERCV INT** – устанавливается в «1», когда размер принимаемого пакета превышает значение, установленное в регистре ERCV TRESHOLD (банк 3, смещение 0xC).

**EPH INT** – устанавливается в «1», когда секция «протокол» в заголовке пакета не соответствует заданному значению.

**RX\_OVRN INT** – устанавливается в нескольких случаях: нет места в приемном буфере, пакет слишком большой, установлен бит RCV DISCRD в регистре ERCV.

**ALLOC INT** – успешное выполнение команды выделения памяти для передачи (MMU).

**TX\_EMPTY INT** – очередь пакетов на передачу пуста.

**TX INT** – устанавливается если пакет успешно передан или в случае ряда ошибок: TXUNRN, SQET, LOST CARR, LAT COL, 16COL.

**RCV INT** – принят пакет.

### Банк 3, регистр MMI интерфейса

HIGH BYTE	Reserved	MSK_ CRS100	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	0	0	1	1	0	0	1	1
LOW BYTE	Reserved				MDOE	MCLK	MDI	MDO
	0	0	1	1	0	0	MDI Pin	0

**MDO** – управление выводом передачи.

**MDI** – вход.

**MCLK** – вывод синхронизации.

**MDOE** – вывод управления передачей (Output Enable).

## Инициализация контроллера LAN91C111

### Сброс контроллера

Сброс производится установкой бита RESET контроллера в логическую «1» на время большее 100 нс. Во время сброса все регистры контроллера переводятся в исходное состояние указанное в документации.

### Инициализация блока PHY

При инициализации блока PHY необходимо включить внутренний блок PHY путем записи соответствующего значения в регистр конфигурации.

### Инициализация приема и передачи

Необходимо инициализировать регистры управления приемом и передачей, а также регистр управления приемом физического уровня.

```
/**-----
                                init_eth
-----
Инициализация контроллера ethernet

Вход:      mode - режим работы драйвера
Выход:     нет
Результат: нет
Описание:
Пример:
-----*/

void init_eth( unsigned long mode )
{
    volatile long i;

    // Сброс контроллера
    IO3SET_bit.P3_29 = 1;

    for( i = 0; i < 50000; i++ );    // задержка

    IO3CLR_bit.P3_29 = 1;

    // Инициализация PHY
    init_eth_phy();

    for( i = 0; i < 50000; i++ );    // задержка

    write_phy_reg( PHY_CONTROL, 0x3000 );

    ETH_BANK = 0; ETH_RCR      = 0x0102;
    ETH_BANK = 0; ETH_TCR      = 0x0001;
    ETH_BANK = 0; ETH_RPCR     = 0x2800;
}
```

### Чтение пакета

```
int read_eth_pack( unsigned char * buf, unsigned short * len )
{
    unsigned short wtmp, n, i, k;
    unsigned short interrupt_status, fifo_ports;

    ETH_BANK = 2; interrupt_status = ETH_INTERRUPT;
    ETH_BANK = 2; fifo_ports       = ETH_FIFO_PORTS;
```

```
if( ( interrupt_status & 0x0001 ) &&
    !( fifo_ports      & 0x8000 ) )
{
    ETH_BANK = 2; ETH_POINTER = 0xE000;
    ETH_BANK = 2; n          = ETH_DATA0 & 0x7FF;

    n -= 4;

    *len = n;

    if( n%2 )
        n = n >> 1 + 1;
    else
        n = n >> 1;

    k = 0;

    for( i = 0; i < n; i++ )
    {
        ETH_BANK = 2; wtmp = ETH_DATA0;

        buf[k++] = wtmp;
        buf[k++] = wtmp >> 8;
    }

    ETH_BANK = 2; ETH_MMU_COMMAND = 0x0080; // Убираем переданный пакет из очереди

    return OK;
}
else
    return ERROR;
}
```

## Запись пакета

```
int write_eth_pack( unsigned char * buf, unsigned short len )
{
    unsigned short wtmp, n, i, k, busy;
    unsigned short interrupt_status, allocate_number;
    unsigned long t1;

    ETH_BANK = 2; ETH_MMU_COMMAND = 0x0020; // выделяем память для пакета

    // Ждем когда память под пакет выделится

    t1 = clock();

    for( ;; )
    {
        ETH_BANK = 2; interrupt_status = ETH_INTERRUPT;

        if( interrupt_status & 0x0008 ) break;

        if( dtimer( t1 ) > ETH_ALLOCATE_TO )
        {
            #if ETH_DEBUG
                printf("write_eth_pack(): error of allocation memory for transmitting\n");
            #endif
            return ERROR;
        }
    }

    ETH_BANK = 2; allocate_number = ETH_PNR >> 8;
    ETH_BANK = 2; ETH_PNR        = allocate_number;

    ETH_BANK = 2; ETH_POINTER      = 0x4000;

    ETH_BANK = 2; ETH_DATA0 = 0;          // status
    ETH_BANK = 2; ETH_DATA0 = len + 6;    // packet len

    n = len;

    if( n%2 )
        n = n >> 1 + 1;
    else
```

```
        n = n >> 1;

        k = 0;

        for( i = 0; i < n; i++ )
        {

            wtmp = buf[ k + 1 ]; wtmp <= 8;
            wtmp += buf[ k ];

            k += 2;

            ETH_BANK = 2; ETH_DATA0 = wtmp;

        }

        ETH_BANK = 2; ETH_TCR = 0x0001; // TXENA = 1;

// Ждем, когда флаг Busy установится в 0

        while( 1 )
        {
            ETH_BANK = 2; busy = ETH_MMU_COMMAND;

            if( busy & 0x0001 ) break;

            if( dtime( t1 ) > ETH_TRANSMIT_TO )
            {
#if ETH_DEBUG
                printf("write_eth_pack(): TO of transmitting\n");
#endif
                return ERROR;
            }
        }

// Сбрасываем последний переданный пакет

        ETH_BANK = 2; ETH_MMU_COMMAND = 0x00C0;

        return OK;
    }
```

## Литература

1. Учебный стенд SDK-2.0. Инструкция по эксплуатации
2. Ю.А. Семенов. Телекоммуникационные технологии. URL: <http://book.itep.ru/>
3. Описание технологии Fast Ethernet. URL: <http://www.ixbt.com>
4. Лаем Куин, Ричард Рассел "Fast Ethernet"
5. К. Заклер "Компьютерные сети";
6. В.Г. и Н.А. Олифер "Компьютерные сети"
7. LPC2119/2129/2194/2292/2294 USER MANUAL. Philips Semiconductors.
8. SMSC LAN91C111 Datasheet
9. SMSC LAN91C111 Application Note 9.6
10. SMSC LAN91C111 FAQ