

Окончание. Начало в № 5'2003

Контроллеры Fast Ethernet для встроенных применений

Иосиф Каршенбойм

lk@mail.loniis.spb.su

Описание блоков FIFO в канале приема и передачи данных

Для различных реализаций проекта могут потребоваться различные варианты исполнения FIFO, поэтому блоки полностью параметризованы и имеют следующие параметры:

- Data_width = 16 — шина данных на запись, разрядность;
- Ctrl_width = 16 — шина служебных слов на запись, разрядность;
- Frame_min_len = 8 — длина минимального пакета в словах Data_width;
- Fifo_len = 128 — «глубина» буфера в словах Data_width.

Рассмотрим более подробно FIFO на прием

На рис. 1 приведено изображение блока FIFO на прием (функции gx_fifo), а сигналы этого блока перечислены в таблице 1.

Приемный FIFO на стороне записи

Приемный FIFO на стороне записи все сигналы получает из приемной части блока mii2_16. Синхрочастота для записи (wrclk) приходит из трансивера, причем трансивер выбирает частоту после стабилизации работы линии связи. Шина данных на запись в FIFO wrData[Data_width-1... 0] синхронизируется сигналом разрешения записи данных (wr_data_ena), который также приходит из mii2_16 и устанавливается, когда принято слово данных полной разрядности 16 или 32 бита, в зависимости от разрядности шины FIFO. Слово управления

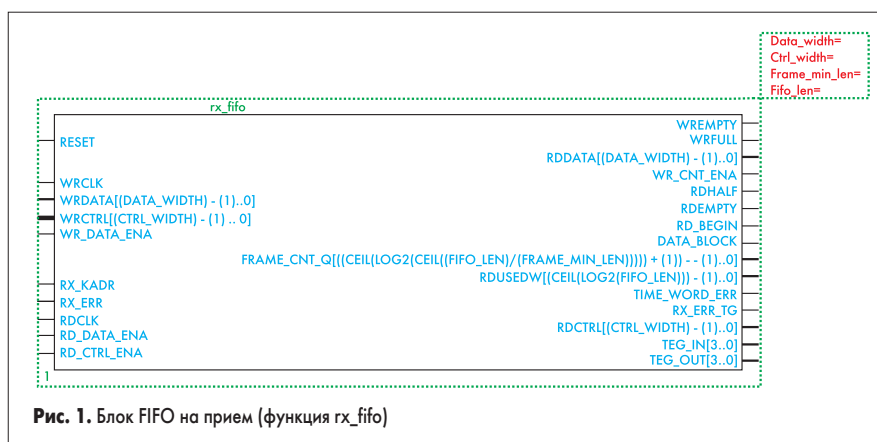


Таблица 1. Сигналы блока FIFO на прием

Группа сигналов	Название сигнала	Направление	Описание
Common	reset	INPUT	вход системного сброса
FIFO на стороне записи	wrclk	INPUT	частота для записи
	wrData [Data_width-1... 0]	INPUT	шина данных на запись
	wr_data_ena	INPUT	разрешение записи данных
	wrCtrl [Ctrl_width-1... 0]	INPUT	шина управления: счетчик принятых слов (байт), ошибки, адреса и пр. НА ЗАПИСЬ
	rx_kadr	INPUT	можно принимать кадр (строб кадра)
	rx_err	INPUT	когда в потоке данных пришло gx_err
	wrempty	OUTPUT	сигнал о пустоте FIFO
	wrfull	OUTPUT	переполнение FIFO
FIFO на стороне чтения	rdclk	INPUT	частота для чтения
	rd_data_ena	INPUT	разрешение чтения данных
	rd_ctrl_ena	INPUT	разрешение чтения регистра управления
выходы управления на стороне записи	rdData [Data_width-1... 0]	OUTPUT	шина данных на чтение
	wr_cnt_ena	OUTPUT	разрешение на запись состояния счетчика принятых слов
выходы управления на стороне чтения	rdhalf	OUTPUT	заполненность данных на половину
	rdempty	OUTPUT	сигнал об отсутствии записей в FIFO
	rd_begin	OUTPUT	в FIFO есть хотя бы один целый кадр или он заполнен на половину, можно использовать как запрос на прерывание «по уровню» для начала чтения буфера
	data_block	OUTPUT	можно использовать как запрос на прерывание «по уровню» для окончания чтения буфера
	frame_cnt_q [FRAME_WIDTH-1... 0]	OUTPUT	счетчик числа целых пакетов, находящихся в буфере
	rdusedw [FIFO_WIDTH-1... 0]	OUTPUT	в FIFO есть данные
выходы ошибок на стороне записи	time_word_err	OUTPUT	когда слово данных пришло, а буфер по записи полон
	rx_err_tg	OUTPUT	когда в потоке данных пришло gx_err
выходы регистра контроля на стороне чтения	rdCtrl [Ctrl_width-1... 0]	OUTPUT	регистр контроля: счетчик принятых слов (байт), ошибки, адреса и пр. — на чтение
выходы для отладки управления	teg_in[3... 0]	OUTPUT	
	teg_out[3... 0]	OUTPUT	

(счетчик принятых слов, ошибки, адреса и пр.) записывается в FIFO со входов шины управления wrCtrl[Ctrl_width-1... 0].

Управляющий сигнал rx_kadr сообщает блоку FIFO, что можно принимать кадр (строб кадра). Этот сигнал устанавливается в случае, если MAC-адрес входящего из сети пакета совпал с MAC-адресом контроллера, либо если входящий пакет — широковещательный. Блок mii2_16 установит строб кадра при выдаче первого слова данных в FIFO.

Сигнал rx_err передает информацию о том, что в потоке данных пришли коды ошибки и этот пакет обрабатывать на верхнем уровне не нужно. На основании обработки состояния FIFO формируются сигналы time_word_err (когда слово данных пришло, а буфер записи полон) и rx_err_tg (когда в потоке данных пришел сигнал rx_err).

Сигналы wrempty (сигнал о пустоте FIFO) и wrfull (переполнение FIFO) используются для проверки режимов работы FIFO. Сигнал wr_cnt_ena — разрешение на запись состояния счетчика принятых слов в регистр контроля.

Приемный FIFO на стороне чтения

Приемный FIFO на стороне чтения работает под управлением хоста. Сигнал rdclk (частота для чтения) — это системная частота хоста, сигналы rd_data_ena (разрешение чтения данных) и rd_ctrl_ena (разрешение чтения регистра управления) — это дешифрованные адреса регистров чтения со стороны хоста. Шина данных на чтение rdData[Data_width-1... 0] поступает из FIFO в хост.

На стороне чтения так же формируются сигналы о заполнении FIFO — rdhalf (заполнен наполовину) и rdempty (отсутствие записей в FIFO). Сигнал rd_begin показывает, что в FIFO есть хотя бы один целый кадр или что он заполнен наполовину — его можно использовать как запрос на прерывание по уровню для начала чтения буфера. Сигнал data_block показывает, что кадр из FIFO считан весь, и дальнейшее чтение данных из FIFO заблокировано — его можно использовать как запрос на прерывание по уровню для окончания чтения буфера.

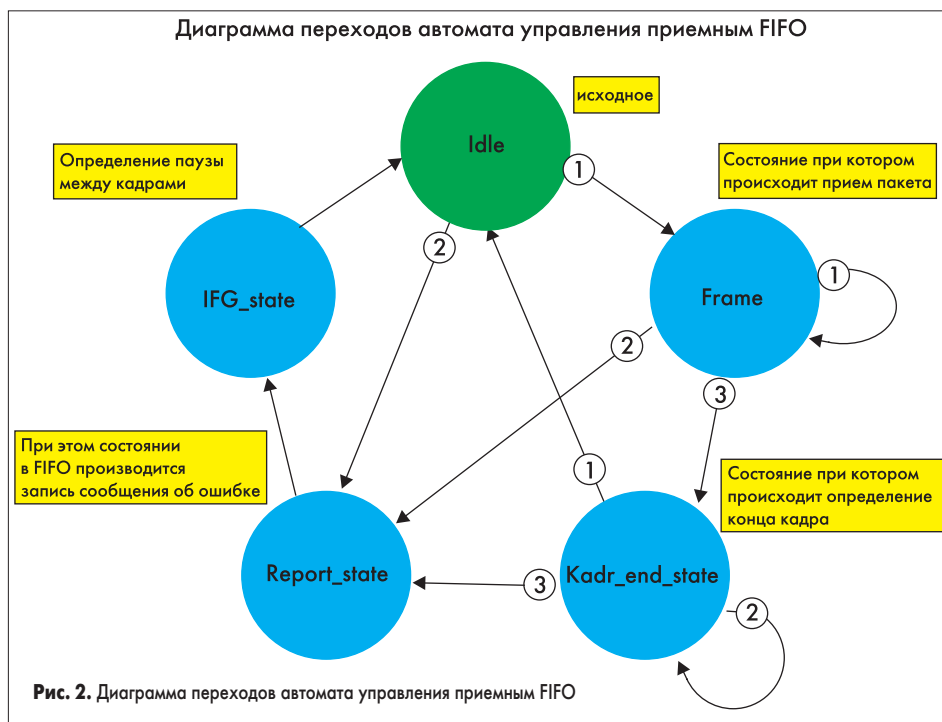
Счетчик числа целых пакетов, находящихся в Буфере — frame_cnt_q[FRAME_WIDTH-1... 0].

Работа приемного FIFO

Блок mii2_16 устанавливает строб кадра и выдает слова данных, которые записываются в FIFO.

При необходимости в буфер помещаются служебные слова, причем их число не ограничено. Единственное ограничение — слова данных во время приема кадра имеют приоритет над записью служебных слов.

Последнее слово данных в буфере не имеет признака конца кадра, так как этот признак может быть выработан только после приема последнего слова по снятию строба кадра. Поэтому следом за последним словом данных должно быть записано служебное



слово с признаком конца кадра. Это слово должно содержать информацию о числе слов в пакете, ошибках, адресации и пр.

Чтение данных из приемного FIFO производится либо «порциями» фиксированной длины, например 16 слов, либо до получения прерывания, сигнализирующего о конце пакета. Чтобы не прочесть «лишних» данных, автомат управления приемным FIFO блокирует чтение слов данных после получения признака конца кадра. Блокировка будет снята только при чтении слова состояния по адресу регистра статуса.

Биты ошибок сбрасываются при чтении по адресу «wr_err_ena». Диаграмма работы автомата управления приемным FIFO представлена на рис. 2 и рис. 7 из первой части статьи.

Автомат управления приемным FIFO rx_process имеет следующие состояния:

- Idle — исходное;
- Frame — состояние, при котором происходит прием пакета;
- Kadr_end_state — состояние, при котором происходит определение конца кадра;
- Report_state — при этом состоянии в FIFO производится запись сообщения об ошибке;
- IFG_state — определение паузы между кадрами.

До начала приема данных автомат управления приемным FIFO находился в состоянии Idle.

Когда блок mii2_16 установит строб кадра и выдает первое слово данных в FIFO, происходит следующее:

1. Если буфер пуст, то автомат управления приемным FIFO, получив признак начала кадра, помещает первое слово данных в буфер с признаком «начала кадра» и затем переходит в состояние Frame.
2. Если буфер полон, то есть хост не считал предыдущий кадр, то автомат управления приемным FIFO переходит в состояние Report_state, в котором он ждет освобождения буфера и по освобождении помещает туда сообщение об ошибке. Затем автомат

переходит в состояние IFG_state и ждет окончания текущего кадра путем определения паузы между кадрами, после чего переходит в исходное состояние — Idle.

Находясь в состоянии Frame, автомат управления приемным FIFO может осуществить три перехода:

1. Если буфер не полон, есть rx_kadr (строб кадра), то есть можно принимать кадр и есть сигнал wr_data_ena (разрешение записи данных), то производится запись данных в FIFO. Автомат управления продолжает находиться в состоянии Frame. Это рабочая ветвь по приему данных.
2. Если есть rx_kadr (строб кадра), то есть можно принимать кадр и есть сигнал wr_data_ena, но буфер полон, то произвести запись данных в FIFO невозможно и слово данных будет потеряно. Поэтому автомат переходит в состояние Report_state, в котором он ждет освобождения буфера и по освобождении помещает туда сообщение об ошибке. Затем автомат переходит в состояние IFG_state и ждет окончания текущего кадра путем определения паузы между кадрами, после чего переходит в исходное состояние — Idle.
3. Если сигнал rx_kadr (строб кадра) снимается, то это значит, что последнее принятое слово данных было последним в этом кадре, кадр закончился и можно переходить в состояние Kadr_end_state, а оттуда — в Idle. Находясь в состоянии Kadr_end_state, автомат управления приемным FIFO может осуществить три перехода:
 1. По нормальному завершению — в Idle.
 2. Если буфер полон, но следующий кадр еще не пришел, то в Kadr_end_state, здесь производится ожидание освобождения буфера.
 3. Если буфер полон, но следующий кадр уже пришел, то производится переход в состояние Report_state, в котором автомат ждет освобождения буфера и по освобождении помещает туда сообщение об ошибке.

Затем автомат переходит в состояние IFG_state и ждет окончания текущего кадра путем определения паузы между кадрами, после чего переходит в исходное состояние — Idle.

Находясь в состоянии IFG_state, автомат управления приемным FIFO может осуществить только один переход — по определению окончания кадра — в состояние Idle.

Кроме автомата управления, в состав приемного FIFO так же входит счетчик числа целых пакетов, находящихся в буфере. Счетчик инкрементирует свое значение при записи слова управления «конца» пакета на стороне записи, то есть когда в буфер помещен целый кадр и сопровождающее его слово управления. Счетчик декрементирует свое значение при прочтении одного кадра на стороне чтения. Сигнал rd_begin означает, что в FIFO есть хотя бы один целый кадр, то есть значение счетчика числа целых пакетов, находящихся в буфере не равно нулю или FIFO заполнен на половину. Этот сигнал можно использовать как запрос на прерывание по уровню для начала чтения буфера.

Как было сказано выше, данный блок поддерживает единую структуру сигналов управления. Сигнал gx_kadr от нижнего по иерархии блока запускает все процессы в данном блоке. Обработав данные, блок приемного FIFO передает верхнему по иерархии блоку сигнал rd_begin, который в свою очередь запускает процессы чтения данных из FIFO в верхнем блоке.

Работа FIFO на передачу

На рис. 3 приведено изображение блока FIFO на передачу — функции tx_fifo, а сигналы этого блока перечислены в таблице 2.

Передающий FIFO получает данные и сигналы управления от хоста, поэтому все, что относится к стороне записи — связано с хостом. Wrclk — системная частота хоста, wrData[Data_width-1..0] — шина данных на запись от хоста, все сигналы вида wr_xh_ena — это продешифрованные сигналы разрешения записи, которые формируются на дешифраторе адресов регистров, поступающих от хоста. Что же касается сигналов, которые формирует сам FIFO, то они тоже поступают в хост. Это сигналы состояния: wrempty — сигнал о пустоте в FIFO, wrfull — переполнение FIFO и wrhalf — заполненность данных на половину. Кроме того, в хост поступают сигналы от счетчика числа целых пакетов, находящихся в буфере, и от триггеров ошибок, которые проверяют правильность работы с FIFO.

Передающий FIFO передает данные и сигналы управления в передающую часть блока mii2_16, поэтому все, что относится к стороне чтения, связано с трактом передачи в линию. Rdclk — частота для чтения в передающем FIFO — формируется в трансивере после того, как он определит режим работы линии. RdData[Data_width-1..0] — шина данных на чтение из FIFO — поступает в передающую часть блока mii2_16, где из слов данных формируются 4-битные nibblы для пе-

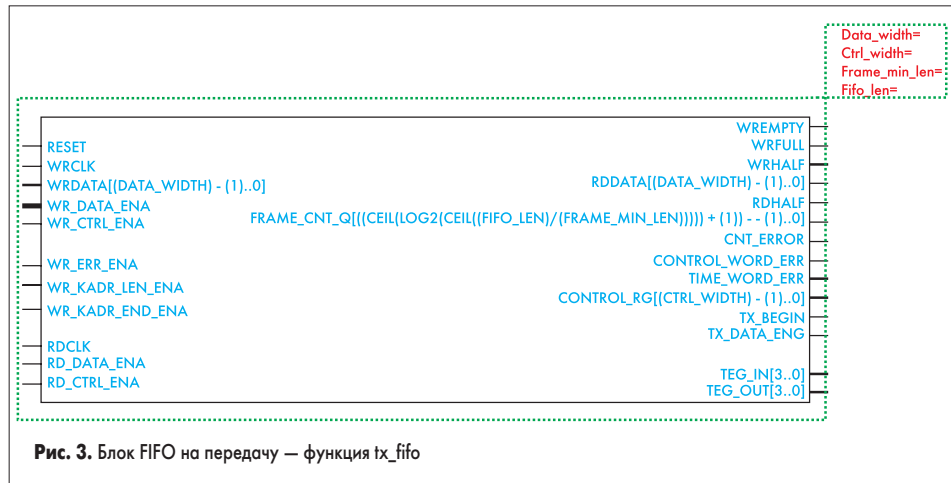


Рис. 3. Блок FIFO на передачу — функция tx_fifo

редачи. Чтение данных производится по сигналам rd_data_ena — разрешение чтения данных, которые формируются в блоке mii2_16 во время передачи данных. Так же, на сторону чтения передаются и сигналы «сопровождения»: tx_begin — признак того, что можно передавать кадр и tx_data_end — признак окончания кадра.

Передающий FIFO работает следующим образом:

1. По адресу регистра начала кадра (wr_kadr_len_ena) хост записывает 16-битное слово, означающее XXXX — число слов в пакете. Запись этого слова трактуется как начало кадра.
2. По адресу регистра данных — wr_data_ena записываются слова данных, их число определяется емкостью буфера и величиной кадра. Если весь кадр не поместился в буфер, то он будет дописан при частичном освобождении буфера.
3. Последнее слово кадра записывается по адресу регистра конца кадра — wr_kadr_end_ena. Туда можно записывать и «пустое» слово, но оно также будет передано в кадре.
4. По адресу регистра управления wr_ctrl_ena может быть записана любая служебная информация, которая тоже будет помещена в тот же кадр данных с признаком служебной информации и далее будет извлечена из пакета при чтении. Можно записывать не более 3-х слов служебной информации подряд для режима 100 мегабит, кроме случая, когда требуется прервать передачу. В противном случае будет взведен триггер control_word_err — ошибка в положении слова управления в пакете.
5. Чтение данных из FIFO на стороне приема начинается в том случае, если записан целый пакет, или буфер заполнен на половину.
6. На приемной стороне производится контроль числа слов в пакете, и если есть несоответствие, то в поток данных, передаваемых далее в МП, вставляется сигнал «ошибка передачи» и взводится триггер ошибки длины пакета.
7. Если чтение данных из FIFO уже началось, и FIFO уже освободился, а новые данные не дописаны, то есть нет признака окончания кадра, то в поток данных вставляется сигнал «ошибка передачи» и взводится триггер time_word_err — когда из буфера все считано, а слово с «концом» пакета не пришло.

Таблица 2. Сигналы блока FIFO на передачу

Группа сигналов	Название сигнала	Направление	Описание
Common	reset	INPUT	вход системного сброса
FIFO на стороне записи	wrclk	INPUT	частота для записи
	wrData [Data_width-1..0]	INPUT	шина данных на запись
	wr_data_ena	INPUT	разрешение записи данных
	wr_ctrl_ena	INPUT	разрешение записи слова управления
	wr_err_ena	INPUT	разрешение чтения триггера ошибки
	wr_kadr_len_ena	INPUT	разрешение записи слова длины пакета
	wr_kadr_end_ena	INPUT	разрешение записи слова «конца» пакета
	wrempty	OUTPUT	сигнал о пустоте FIFO данных
	wrfull	OUTPUT	переполнение FIFO данных
FIFO на стороне чтения	wrhalf	OUTPUT	заполненность данных на половину
	rdclk	INPUT	частота для чтения
	rd_data_ena	INPUT	разрешение чтения данных
	rd_ctrl_ena	INPUT	разрешение чтения
	rdData [Data_width-1..0]	OUTPUT	шина данных на чтение
выходы управления на стороне записи	rdhalf	OUTPUT	заполненность данных на половину
	frame_cnt_q [FRAME_WIDTH-1..0]	OUTPUT	счетчик числа целых пакетов, находящихся в Буфере
	cnt_error	OUTPUT	триггер ошибки длины пакета
выходы ошибок на стороне записи	control_word_err	OUTPUT	ошибка в положении слова управления в пакете
	time_word_err	OUTPUT	когда из буфера все считано, а слово с «концом пакета» не пришло
выходы управления на стороне чтения	control_rg [Ctrl_width-1..0]	OUTPUT	выходная шина регистра контроля
	tx_begin	OUTPUT	можно передавать кадр
выходы для отладки управления	tx_data_end	OUTPUT	данные закончились
	teg_in[3..0]	OUTPUT	
	teg_out[3..0]	OUTPUT	

Автомат управления передающим FIFO имеет два состояния: Idle и Frame, см. первую часть статьи, рис. 7.

На стороне чтения производится определение состояния выходов тегов. Автомат ждет появления кода начала пакета и нахо-

дится в состоянии Idle. При поступлении кода, соответствующего признаку начала пакета, автомат переходит в состояние Frame. При этом производится чтение слова из FIFO и значение, соответствующее числу последующих слов в пакете помещается в счетчик, который будет проверять количество переданных слов. Далее в состоянии Frame возможны следующие события:

1. Следующее слово на выходе FIFO — слово управления. Автомат управления формирует импульс чтения слова из FIFO. Слово управления считывается и помещается в регистр управления. Автомат управления продолжает находиться в состоянии Frame. Сигнал разрешения передачи для блока mii2_16 — tx_begin на этом этапе еще не формируется.
2. На выходе FIFO нет данных. Автомат управления продолжает находиться в состоянии Frame.
3. На выходе FIFO есть данные. Автомат управления продолжает находиться в состоянии Frame. Если на стороне записи в FIFO был записан признак «конца пакета», то счетчик пакетов, находящихся в FIFO, инкрементируется, и этот сигнал поступает в автомат управления. Он формирует сигнал разрешения передачи tx_begin для блока mii2_16. Чтение данных производится по сигналам разрешения чтения данных rd_data_ena, поступающим от mii2_16.
4. На выходе FIFO есть данные. Автомат управления продолжает находиться в состоянии Frame. Если на стороне записи в FIFO записаны данные в FIFO установится признак wrhalf — заполненность FIFO на половину, но признак «конца пакета» еще не записан, то счетчик пакетов, находящихся в FIFO не инкрементируется. То есть началась передача большого пакета, и он целиком не помещается в FIFO. Поэтому сигнал wrhalf тоже поступает в автомат управления. Он формирует сигнал разрешения передачи tx_begin для блока mii2_16. Чтение данных производится по сигналам rd_data_ena.
5. На выходе FIFO есть признак «конец пакета» и счетчик, который проверял количество переданных слов, не показывает ошибки по числу переданных слов. Есть успешное окончание передачи пакета. Счетчик пакетов, находящихся в FIFO, декрементируется, и этот сигнал поступает в автомат управления и в хост. Автомат управления переходит в состояние Idle.
6. На выходе FIFO есть признак «конец пакета», но счетчик, который проверял количество переданных слов, показывает, что пакет передан не весь и данные «кончились рано». Возводится триггер ошибки длины пакета cnt_error. Происходит завершение передачи пакета. Счетчик пакетов, находящихся в FIFO, декрементируется, и этот сигнал поступает в автомат управления и в хост. Автомат управления переходит в состояние Idle.
7. На выходе FIFO нет признака «конец пакета», но счетчик, который проверял количество переданных слов, показывает, что пакет передан весь. Возводится триггер ошиб-

ки длины пакета cnt_error. Происходит завершение передачи пакета. Счетчик пакетов, находящихся в FIFO, декрементируется, и этот сигнал поступает в автомат управления и в хост. Автомат управления переходит в состояние Idle.

8. На выходе FIFO нет данных и FIFO пуст, но счетчик, который проверял количество переданных слов, показывает, что пакет передан не весь. Формируется сигнал ошибки time_word_err — когда из буфера все считано, а следующее слово не пришло. Ситуация возможна, если хост не успел дописать в FIFO очередное слово. По этому состоянию взводится триггер ошибки и блок mii2_16 устанавливает сигнал tx_er, по которому трансивер вставляет в поток данных на передачу соответствующие коды и сообщает абоненту о том, что в передаваемом потоке данных произошла ошибка. Автомат управления продолжает находиться в состоянии Frame.
9. На выходе FIFO находится слово управления, но его не успели считать из FIFO к моменту прихода сигнала rd_data_ena — разрешение чтения данных. Формируется сигнал control_word_err — ошибка в положении слова управления в пакете. Ситуация возможна, если хост записал в FIFO слов управления больше, чем можно успеть считать. По этому состоянию взводится триггер ошибки и блок mii2_16 устанавливает сигнал tx_er, по которому трансивер вставляет в поток данных на передачу соответствующие коды и сообщает абоненту о том, что в передаваемом потоке данных произошла ошибка. Автомат управления продолжает находиться в состоянии Frame. По окончании передачи, когда из FIFO извлекается признак конца кадра, что происходит вместе чтением последнего слова данных, в блок mii2_16 выдается сигнал tx_data_end о том, что передача данных текущего кадра закончена.

Описание блока mii2_16

Блок mii2_16 и является тем самым блоком, который выполняет основные функции MAC-контроллера. Он преобразует интерфейс FIFO в интерфейс MII, формирует передаваемый кадр данных, производит подсчет CRC на стороне передачи и автоматически добавляет CRC к кадру данных на передаче. Кроме того, на стороне передачи автоматически формируется пауза между кадрами — IFG. На стороне приема блок mii2_16 производит прием кадров от трансивера, выделяет кадры, предназначенные для данного контроллера или широковещательные кадры, производит проверку кадра по CRC. На стороне приема блок mii2_16 так же производит согласование интерфейса MII и интерфейса FIFO. Блок формирует 16 или 32-битные слова из принимаемых байтов, причем по окончании кадра блок дополняет последнее принятое слово до требуемой разрядности.

На рис. 4 приведено изображение блока mii2_16, а сигналы этого блока перечислены в таблице 3.

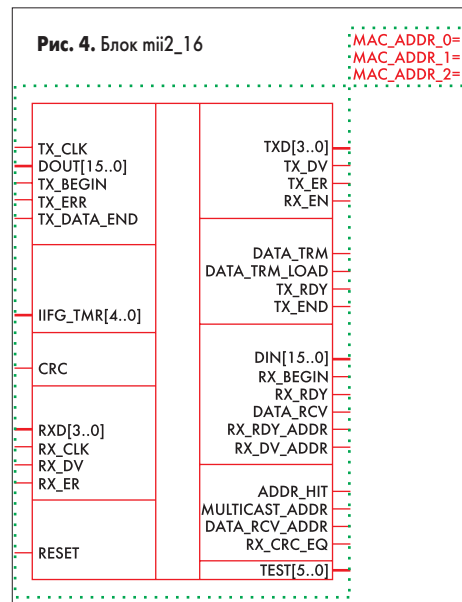


Таблица 3. Сигналы блока mii2_16

Группа сигналов	Название сигнала	Направление	Описание
Common	reset	INPUT	вход системного сброса, активный 1
Управление передатчиком	Dout[15..0]	INPUT	параллельная шина данных из процессора 1 – вызывает начало передачи, выдается под фронт TX_CLK
	data_trm	OUTPUT	1 – идет передача данных, преамбула уже передана
	data_trm_load	OUTPUT	1 – идет загрузка данных в передатчик
	tx_rdy	OUTPUT	1 – «слово передано»
	tx_err	INPUT	активный 1 – выдается под фронт TX_CLK
	tx_data_end	INPUT	активный 1 – конец передачи данных, далее CRC, выдается под фронт TX_CLK
	tx_end	OUTPUT	1 – передача окончена
Управление приемником	ifg_tmr[4..0]	INPUT	загружается длительность таймера, формирующего сигнал IFG
	Din[15..0]	OUTPUT	выходная шина данных – вход для процессора
	rx_rdy	OUTPUT	1 – готов
	data_rcv	OUTPUT	1 – вызывает начало приема данных, выдается под фронт RX_CLK
	rx_rdy_addr	OUTPUT	1 – готов – с учетом определения MAC-адреса
	rx_dv_addr	OUTPUT	Data Valid – с учетом определения MAC-адреса
	addr_hit	OUTPUT	MAC-адрес выбран
	multicast_addr	OUTPUT	есть признак широковещательного кадра
	data_rcv_addr	OUTPUT	1 – вызывает начало приема данных, выдается под фронт RX_CLK
	rx_begin	OUTPUT	с учетом определения MAC-адреса
rx_crc_eq	OUTPUT	1 – подсчет CRC совпал с заданной константой, выдается под фронт RX_CLK	
Интерфейс MII	CRS	INPUT	Carrier Sense indication
Receive DATA	RXD[3..0]	INPUT	Parallel data input (output for transceiver) path
	RX_CLK	INPUT	Recovered clock input (output for transceiver)
	RX_DV	INPUT	Data Valid
	RX_ER	INPUT	Error
Transmit DATA	TXD[3..0]	OUTPUT	Parallel data output (input for transceiver) path
	TX_CLK	INPUT	Recovered clock input (output for transceiver)
	TX_DV	OUTPUT	Data Valid
	TX_ER	OUTPUT	Error

Описание сигналов блока mii2_16.

Блок mii2_16 имеет параметры:

- MAC_ADDR_0 = Н»1234» — записывать адрес в прямом порядке следования байтов
- MAC_ADDR_1 = Н»5678» — например слово 693E писать как 693E
- MAC_ADDR_2 = Н»9ABC»

Блок состоит из двух независимых частей: передающей и приемной.

Блок передатчика содержит следующие узлы:

1. Сдвиговые регистры передаваемых данных
 $tx_data_shift_rg_a[3..0]$,
 $tx_data_shift_rg_b[3..0]$,
 $tx_data_shift_rg_c[3..0]$,
 $tx_data_shift_rg_d[3..0]$.
2. Счетчик nibблов при передаче: tx_nibble_cnt (разрядность 5), выполняющий роль счетчика nibблов и таймера, определяющего паузы.
3. Блок генерации CRC для передачи.

Передающая часть блока mii2_16 получает из FIFO на передачу следующие сигналы:

1. $Dout[15..0]$ — выходная шина данных из FIFO.
2. Сигнал tx_begin — вызывает начало передачи в линию, устанавливается автоматом управления, когда FIFO на передачу заполнен наполовину или в FIFO уже есть один целый кадр.
3. Сигнал tx_err — устанавливается автоматом управления FIFO, когда в потоке данных, поступающих из FIFO по каналу передачи, обнаруживается ошибка.
4. Сигнал tx_data_end — устанавливается автоматом управления FIFO, передача данных данного кадра окончена.

Передающая часть блока mii2_16 отдает в FIFO на передачу следующие сигналы:

1. Сигнал $data_trm$ — устанавливается автоматом управления $tx_process$ и показывает, что идет передача данных, преамбула уже передана.
2. Сигнал $data_trm_load$ — устанавливается автоматом управления $tx_process$ и показывает, что идет загрузка данных в передатчик.
3. Сигнал tx_rdy — устанавливается автоматом управления $tx_process$ и показывает, что «слово передано».
4. Сигнал tx_end — устанавливается автоматом управления $tx_process$ и показывает, что передача кадра полностью завершена.

Передающая часть блока mii2_16 получает из хоста следующие сигналы:

Сигналы, поступающие по шине $ifg_tmr[4..0]$, определяют длительность паузы между кадрами в линии.

Передающая часть блока mii2_16 отдает на передачу в МП следующие сигналы:

- TXD[3..0]: OUTPUT — Parallel data output (input for transceiver) path;
- TX_CLK: INPUT — Recovered clock input (output for transceiver);
- TX_DV: OUTPUT — Data Valid;
- TX_ER: OUTPUT — Error.

Описание и назначение сигналов МП приведено в описании интерфейса МП.

Процессом передачи управляют автомат управления процессом передачи $tx_process$ и счетчик nibблов при передаче tx_nibble_cnt .

Автомат управления процессом передачи $tx_process$ имеет следующие состояния:

- Idle — исходное;
- Data_state — состояние при котором передаются данные;
- Wait — состояние ожидания;
- Crc_state — состояние при котором производится передача в линию CRC;
- IFG_state — состояние формирования паузы между кадрами в линии;
- Pause_state — ожидания снятия сигнала tx_begin .

Процесс передачи начинается с выдачи из хоста в блок FIFO на передачу одного целого кадра или FIFO на передачу заполняется данными наполовину. При этом автомат управления FIFO выдаст в блок mii2_16 сигнал управления $tx_begin = 1$. Этот сигнал вызывает начало передачи. Сигнал tx_begin запускает счетчик tx_nibble_cnt , а затем — автомат управления процессом передачи. Счетчик tx_nibble_cnt начинает отсчет nibблов, формирующих преамбулу, и в линию передается преамбула. Автомат управления процессом передачи $tx_process$ в это время находится в состоянии Idle. Счетчик tx_nibble_cnt производит отсчет тактов при передаче. До тех пор, пока счетчик не досчитает до 14, в линию будет передаваться преамбула. Когда будет достигнуто состояние $tx_nibble_cnt.q[3..0] = 14$, в линию будет передаваться преамбула и старт передачи. В этом же такте из FIFO загрузится первое слово на передачу. И в этом же такте автомат управления процессом передачи перейдет из состояния Idle в состояние Data_state. Когда будет достигнуто состояние $tx_nibble_cnt.q[3..0] = 15$, в линию начнут передаваться данные. Под сигнал $tx_data_load = 1$ производится загрузка передаваемых данных в сдвиговые регистры $tx_data_shift_rg_a[3..0]$, $tx_data_shift_rg_b[3..0]$, $tx_data_shift_rg_c[3..0]$, $tx_data_shift_rg_d[3..0]$. Передача данных в и из МП обычно осуществляется в 4-битном формате — nibблами. Поэтому, когда сигнал tx_data_load снимается, т. е. становится равным нулю, то производится потетрадный сдвиг данных на выходную шину блока TXD[3..0].

Далее автомат управления формирует сигналы tx_rdy , по которым производится считывание данных из блока FIFO. Эти сигналы будут привязаны к частоте TX_CLK. Частота TX_CLK формируется в трансивере и определяется трансивером после автопереговоров в линии или назначается трансиверу при начале работы.

При выдаче nibблов данных в линию блок генерации CRC для передачи производит подсчет CRC.

Процесс передачи данных продолжается до тех пор, пока из блока FIFO не поступит сигнал tx_data_end , который устанавливается автоматом управления FIFO, когда передача данных данного кадра окончена. По получению этого сигнала автомат управления процессом передачи $tx_process$ переходит из состояния Data_state в промежуточное состояние, формирующее такт задержки для

перехода в состояние Pause_state, а оттуда в состояние Crc_state. Когда автомат управления попадает в состояние Crc_state, данные, полученные при подсчете CRC, выдаются в линию потетрадно.

По окончании передачи CRC, автомат управления из состояния Crc_state переходит в состояние IFG_state. При переходе в состояние IFG_state автомат управления формирует сигнал tx_end , который передается автоматом управления FIFO и подтверждает завершение передачи кадра. При получении этого сигнала автомат управления FIFO снимает свой сигнал управления tx_begin .

Во время состояния IFG_state формируется пауза между кадрами в линии. Для этого, по окончании состояния Crc_state счетчик tx_nibble_cnt обнуляется и начинает считать заново. Код, поступающий по шине $ifg_tmr[4..0]$ из хоста, сравнивается с выходом счетчика tx_nibble_cnt , который используется в данном состоянии как таймер. После выдержки времени автомат управления из состояния IFG_state переходит в состояние Pause_state, где он находится в ожидании снятия сигнала tx_begin , поступившего от автомата управления FIFO. Таким образом, следующая передача кадра возможна только в том случае, когда данный кадр передан полностью и оба автомата, участвующие в передаче кадра — автомат управления передачей FIFO и автомат управления передачей mii2_16 полностью выполнили задачу и оба находятся в исходном состоянии для передачи нового кадра.

Блок приемника содержит следующие узлы:

1. Регистры принятых данных $rx_rg_a[3..0]$, $rx_rg_b[3..0]$, $rx_rg_c[3..0]$, $rx_rg_d[3..0]$;
2. Счетчик nibблов при приеме rx_nibble_cnt (разрядность 2);
3. Счетчик слов MAC-адреса при приеме $addr_word_cnt$ (разрядность 2);
4. Блок проверки CRC на приеме: $rx_crc : crc32_nibble$.

Приемная часть блока mii2_16 получает из трансивера следующие сигналы интерфейса МП:

- RXD[3..0]: INPUT — Parallel data input (output for transceiver) path;
- RX_CLK: INPUT — Recovered clock input (output for transceiver);
- RX_DV: INPUT — Data Valid;
- RX_ER: INPUT — Error.

Приемная часть блока mii2_16 отдает в FIFO на прием следующие сигналы:

1. $Din[15..0]$ — выходная шина данных — вход для FIFO.
2. Сигнал готовности выдачи одного слова данных rx_rdy .
3. Сигнал $data_rcv$ — вызывает начало приема данных, выдается под фронт RX_CLK.
4. Сигнал rx_rdy_addr — готовности выдачи слов, что MAC-адрес уже определен.
5. Сигнал rx_dv_addr — Data Valid — с учетом того, что MAC-адрес уже определен.
6. Сигнал $addr_hit$ — MAC-адрес выбран.
7. Сигнал $multicast_addr$ — есть признак широковещательного кадра.
8. Сигнал $data_rcv_addr$ — вызывает начало приема данных с учетом определения MAC-адреса.

9. Сигнал `gx_begin`.

10. Сигнал `gx_crc_eq` — подсчет CRC совпал с заданной константой.

Процесс приема для блока `mii2_16` начинается с определения состояния сигнала `RX_DV`, поступающего от трансивера. Этот сигнал устанавливается, когда из линии в трансивер приходят коды, определенные как коды данных для трансивера. Трансивер помещает первую тетраду на свою выходную шину `RXD[3..0]`, и устанавливает сигнал `RX_DV`. Эти данные под сигнал приемной синхрочастоты `RX_CLK` записываются во входной регистр приемной части блока `mii2_16`. Частота `RX_CLK` формируется в трансивере и определяется трансивером после автопереговоров в линии или назначается трансиверу при начале работы. Приемник проверяет принятые тетрады на совпадение с кодом «5D», соответствующим Start Frame Delimiter (SFD). Код SFD отделяет преамбулу от остальных полей кодов в кадре. При получении SFD запускается узел, определяющий MAC-адрес устройства. MAC-адрес устройства в данном варианте проекта задается как набор параметров: `MAC_ADDR_0`, `MAC_ADDR_1`, `MAC_ADDR_2`. Определитель MAC-адреса устройства состоит из двух частей. Первая определяет уникальный MAC-адрес. Для этого тетрады, приходящие от трансивера на вход приемника, поочередно сравниваются с соответствующими тетрадами MAC-адреса. Этой цели служит счетчик слов MAC-адреса при приеме — `addr_word_cnt`, мультиплексор, схема сравнения и соответствующий триггер. Вторая часть определяет широковещательный адрес. Когда счетчик принятых слов отсчитает 6 принятых байт, производится прием данных, если MAC-адрес выбран или это широковещательный кадр, или приемник заканчивает работу и ждет, когда сигнал `RX_DV` будет снят, а, затем снова установлен с приходом следующего кадра. Если определитель MAC-адреса дает сигнал `gx_dv_addr` о том, что MAC-адрес выбран, то взводится либо триггер `addr_hit` (уникальный MAC-адрес выбран), либо триггер `multicast_addr` (есть признак широковещательного кадра). Затем устанавливается сигнал `gx_begin`, который поступает на вход приемного FIFO и все слова данных, принятые из MII и преобразованные из тетрад в 16-битные слова, помещаются в FIFO. Слова данных стробируются сигналом `gx_rdy_addr`, который формируется при формировании 16-битного слова. Если процесс приема заканчивается с нечетным числом принятых байт, то блок дополняет нулями последнее принятое слово до 16-битного формата. Во время приема кадра производится проверка принятого и обработанного потока данных в блоке приемного CRC. Если совпадение происходит, то устанавливается сигнал `gx_crc_eq` — подсчет CRC совпал с заданной константой. Этот сигнал поступает в приемный FIFO на шину статуса и записывается в FIFO как служебная информация по окончании приема кадра. Сигналы триггеров `addr_hit` и `multicast_addr` также запи-

сываются в FIFO как служебная информация по окончании приема кадра.

Сигнал `CRS` не обрабатывается в блоке `mii2_16`, поскольку его нет необходимости обрабатывать, когда установлен дуплексный режим.

Описание контроллера MI в FPGA

Контроллер представляет собой блок, формирующий диаграмму интерфейса MII MI, описанные выше, см. первую часть статьи, рис. 6.

На рис. 5 приведено изображение блока `mii`, а сигналы этого блока перечислены в таблице 4.

Блок состоит из частей:

1. Узел связи с интерфейсом хоста — блока регистров, на которых хранится адрес и данные для передачи в MI.
2. Приемо-передающий узел для обмена сигналами по интерфейсу MI на сдвиговом регистре, счетчик битов и автомат управления передатчиком и приемником.

Блок `mii` содержит следующие узлы:

6-разрядный счетчик битов `bit_cnt`, буферный регистр передаваемых адресов `Addr[10..0]`, буферный регистр передаваемых данных `D[15..0]`, сдвиговый регистр принятых и передаваемых данных `shift_rg[15..0]`, триггер готовности `rdy_tg`.

Для упрощения принято, что код операции будет представлять собой старший бит в слове адреса, тогда с точки зрения программы будут отдельные адреса для записи и отдельные для чтения данных.

Порядок работы с блоком `mii` следующий. Сигнал записи в регистр адреса `A_wrn` инициирует процесс обмена по интерфейсу MI. Поэтому для чтения данных из MI нужно только произвести запись слова адреса, которое будет содержать код операции «чтение», адрес требуемого трансивера и адрес регистра трансивера. Во время цикла обмена готовность снимается, а по завершении обмена — устанавливается. При установлении триггера готовности `rdy_tg` на выходной шине блока `Din[15..0]` будут находиться данные, полученные из MI. Для записи слова данных в MI нужно сначала записать слово данных для передачи в MI в регистр передаваемых данных. После этого нужно записать слово адреса, но с кодом операции «запись» и аналогично указать адрес требуемого трансивера и адрес регистра трансивера. Далее, как и в цикле чтения, при установлении триггера готовности `rdy_tg` блок будет готов к следующему циклу обмена.

Блок работает следующим образом. При записи в регистр адреса по положительному перепаду сигнала `A_wrn` триггер готовности сбрасывается и разрешает работать счетчику битов `bit_cnt`, который начинает отсчитывать такты преамбулы. Выходной трехстабильный порт MDIO исходно открыт и в MI выдаются биты преамбулы. Когда счетчик доходит до значения `N»1F»` в сдвиговый регистр `shift_rg[15..0]` переписываются стартовые биты, код операции, код адреса и значения двух битов, требуемых для разворота шины. При записи эти два бита имеют

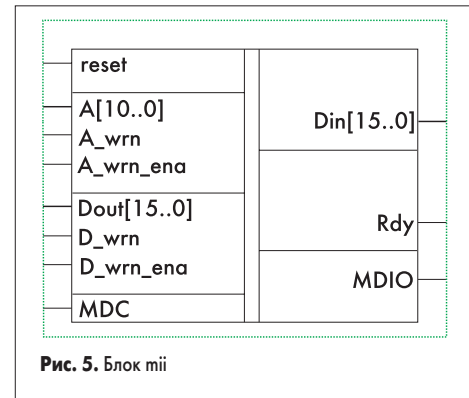


Рис. 5. Блок `mii`

Таблица 4. Сигналы блока `mii`

Группа сигналов	Название сигнала	Направление	Описание
Common	reset	INPUT	вход системного сброса, активный 1
интерфейсные сигналы связи с хостом блока <code>mii</code>	Dout[15..0]	INPUT	шина данных от хоста
	D_wrn	INPUT	0 – Сигнал импульса записи слова данных
	D_wrn_ena	INPUT	1 – Сигнал разрешения импульса записи, это дешифрованный адрес регистра
	A[10]	INPUT	при 1 соответствует коду операции «запись», при 0 – «чтение»
	A[9..5]	INPUT	адрес трансивера
	A[4..0]	INPUT	адрес регистра трансивера
	A_wrn	INPUT	0 – импульс записи слова адреса
	A_wrn_ena	INPUT	1 – разрешение импульса записи, это дешифрованный адрес регистра
по интерфейсу MI блок производит обмен сигналами	Din[15..0]	OUTPUT	выходная шина данных блока – вход для хоста
	rdy	OUTPUT	бит готовности для последующей операции по MI
по интерфейсу MI блок производит обмен сигналами	MDC	INPUT	принимается тактовая частота для MI
	MDIO	BIDIR	двухнаправленной шине данных

значение `B»10»`, а при чтении — `B»00»`. Производится сдвиг, и на выход MDIO выдаются требуемые по интерфейсу значения адреса и служебных битов — старта, кода операции, битов для разворота шины. В цикле записи в MI порт и далее остается открытым, а в цикле чтения, когда счетчик достигает значения `N»2E»`, выходной порт закрывается и остается закрытым до конца цикла обмена. Когда счетчик досчитывает до значения `N»2F»`, в сдвиговый регистр переписываются данные из буферного регистра передаваемых данных `D[15..0]`. Производится цикл приема-передачи данных в MI. Данные с выхода сдвигового регистра поступают на порт с трехстабильным выходом и, если он открыт (цикл записи), то эти данные выводятся в MI. Данные со входа MI поступают на вход сдвигового регистра и заполняют этот регистр. Если производится операция чтения, то к моменту приема данных порт с трехстабильным выходом будет закрыт, и все данные, принимаемые из MDIO, будут данными, которые передавал трансивер. Данные со входа MI заполняют сдвиговый регистр и выдаются на выходную шину данных блока `Din[15..0]`. Эта шина будет входом для хоста при чтении данных из блока. Когда счетчик достигнет

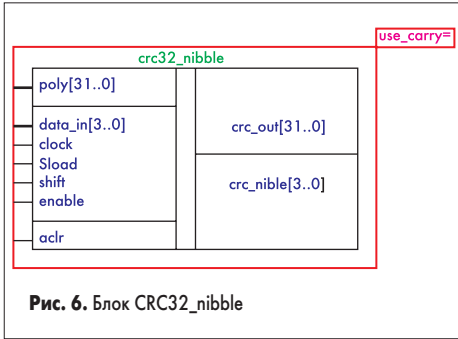


Рис. 6. Блок CRC32_nibble

Таблица 5. Сигналы блока CRC32_nibble

Название сигнала	Направление	Описание
poly[31..0]	INPUT	полином
clock	INPUT	синхрочастота
aclr	INPUT	асинхронный сброс
sload	INPUT	сигнал загрузки исходного состояния
shift	INPUT	сигнал сдвига данных
data_in[3..0]	INPUT	входная шина данных при приеме nibблами
enable	INPUT	сигнал разрешения приема данных
crc_out[31..0]	OUTPUT	выход блока
crc_nibble[3..0]	OUTPUT	выход блока при сдвиге nibблами

значения Н»3F», сформируется сигнал для установки триггера готовности, который в свою очередь, остановит счетчик. Выходной трехстабильный порт MDIO открывает. На этом цикл обмена завершается.

Подсчет CRC32 производится в блоке CRC32_nibble, выполненном в соответствии с документацией Л[5].

Образующий полином: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X^1 + X^0$ преобразуется в константу Н»04C11DB7», подаваемую на вход poly[31..0] блока. Блок представлен на рис. 6, а сигналы блока представлены в таблице 5.

Работа блока

При установке на входе sload = 1 происходит загрузка в регистры исходного состояния для подсчета CRC32.

При sload = 0 и shift = 1, происходит подсчет контрольной суммы. Блок отличается от мегафункции фирмы Altera тем, что если при его использовании задействован вход shift, то при sload = 0 и shift = 0 происходит сдвиг данных из регистра потетрадно.

Программирование MAC-контроллера со стороны хоста

Контроллер состоит из 2-х частей:

- 1. MI — служебный интерфейс
- 2. MAC — контроллер

Программирование MI

- MI предназначен для чтения и записи данных в регистрах трансивера.
 - MI программируется как порт с 16-битным словом.
 - Адресация регистров трансивера косвенная.
1. Для чтения регистра трансивера в MI записывается слово адреса регистра трансивера. Формат слова адреса:

- D [4..0] — register addr — адрес регистра трансивера;
 - D [9..5] — PHY addr — физический адрес самого трансивера, устанавливается переключателями на трансивере;
 - D [15..10] = «0» — чтение.
2. Для записи данных в регистр MI записывается слово адреса регистра трансивера
- Формат слова адреса MI:
- D [4..0] — register addr — адрес регистра трансивера;
 - D [9..5] — PHY addr — физический адрес самого трансивера, устанавливается переключателями на трансивере;
 - D [10] = «1» — запись;
 - D [15..11] = «0».

Чтение и запись производятся по готовности.

Меняя режимы работы трансивера можно устанавливать различные режимы, такие как программный сброс трансивера, состояние петли, позволяющее проверить прохождения данных через MAC-контроллер и трансивер, рабочее состояние. Можно менять режимы индикации светодиодов, подключенных к трансиверу.

Программирование MAC

Обычно уникальный адрес MAC-адрес «прошивается» в устройстве при его изготовлении фирмой-изготовителем. Зоны адресов «поделены» между изготовителями оборудования так, что в сети не должно быть двух устройств с одинаковыми адресами. В рассматриваемом здесь проекте сетевой адрес MAC может задаваться при загрузке FPGA, то есть изготовителем устройства или программно загружаться хост-процессором. В случае загрузки с хост-процессора у сетевого администратора при настройке сети появляется возможность изменить MAC-адрес.

Для примера рассмотрим программирование MAC-контроллера, не имеющего каналов DMA. В такой контроллер данные передаются в программном режиме. MAC-контроллер программируется как порт с 16-битным словом или 32-битным словом, в зависимости от реализации. Данные записываются и читаются либо при опросе готовности FIFO, либо по прерываниям. Сигналы регистра статуса и регистра контроля MAC-контроллера приведены в таблицах 6–7. Эти регистры доступны на чтение и содержат всю информацию, необходимую для контроля работы устройства.

Для записи данных в MAC-контроллер производятся следующие действия:

1. Читается готовность на передачу и, если буфер пуст или пуст наполовину, то в буфер записываются слова данных. Если пакет не влезает в буфер, то ждем освобождения буфера и «подкачиваем» данные порциями до тех пор, пока в буфер не будет записан весь пакет. При этом в буфер дописывается определенное число слов данных, это число, например, может быть равно половине емкости буфера. То же самое производится и в режиме работы по прерываниям.

Таблица 6. Формат регистра статуса MAC

Бит	Название	Описание
0	host_tx_half	буфер по передаче полон на половину
1	host_tx_full	буфер по передаче полон
2	txfifo.cnt_error	триггер ошибки длины пакета
3	txfifo.time_word_err	когда из буфера все считано, а слово с «концом» пакета не пришло
4	host_tx_empty	
5	rxffifo.rd_begin	в FIFO есть хотя бы один целый кадр или оно заполнено на половину, можно использовать как запрос на прерывание по уровню для начала чтения Буфера.-> host_rx_rdy
6	rxffifo.rdempty	сигнал об отсутствии записей в FIFO
7	rxffifo.rdhalf	заполненность данных на половину
8	rxffifo.data_block	можно использовать как запрос на прерывание по уровню для окончания чтения Буфера
9	—	
10	rxffifo.rdctrl[8]	Carrier Sense indication
11	rxffifo.rdctrl[9]	1 – столкновение кадров
12	rxffifo.rdctrl[10]	1 – широкоэвещательный кадр
13	rxffifo.rdctrl[11] # rxffifo.rdctrl[13]	когда слово данных пришло, а буфер по записи полон – 1 – переполнение
14	rxffifo.rdctrl[12]	CRC-ок
15	mii_mi_rdy	готовность служебного канала, 1 – готов

Таблица 7. Формат регистра контроля MAC

Бит	Название	Описание
0..7	rxffifo.rdctrl[]	читаем регистр контроля: счетчик принятых слов (байт), ошибки, адреса и пр. НА ЧТЕНИЕ
8	CRS	
9	COL	
10	rx_multicast_node	широкоэвещательный кадр
11	time_word_err_node	переполнение
12	rx_crc_ok	CRC-ок
13	rx_err_node	RX_ER

2. Когда все слова данных записаны в буфер, то по адресу «конец пакета», записывается еще одно слово и это может быть последнее слово пакета или пустое слово. Обращение хост-процессора по адресу «конец пакета» записывает в FIFO признак «конец пакета» и при этом инкрементируется счетчик числа пакетов на передачу в буфере. Передача пакета инициализируется в двух вариантах. В первом варианте пакет целиком помещается в буфер, и тогда передача начинается при записи признака «конец пакета». Во втором варианте, когда пакет целиком не помещается в буфер, передача начинается при заполнении буфера наполовину.

Для чтения данных из MAC-контроллера производятся следующие действия:

1. Читается готовность на прием, и если готовность есть, то производится чтение данных. Так же на чтение доступен счетчик принятых целых пакетов. То есть, если есть целые пакеты, то читаем целый пакет, иначе читаем только часть пакета (весь пакет не влез в буфер). Чтение данных можно производить пословно, проверяя состояние «буфер пуст», или можно производить чтение данных «порциями» фиксированной длины. При чтении данных в пакетном режиме необходимо проверять состояние «буфер пуст наполовину». При работе с прерываниями чтение можно производить до момента получения прерывания, означающего получение конца кадра,

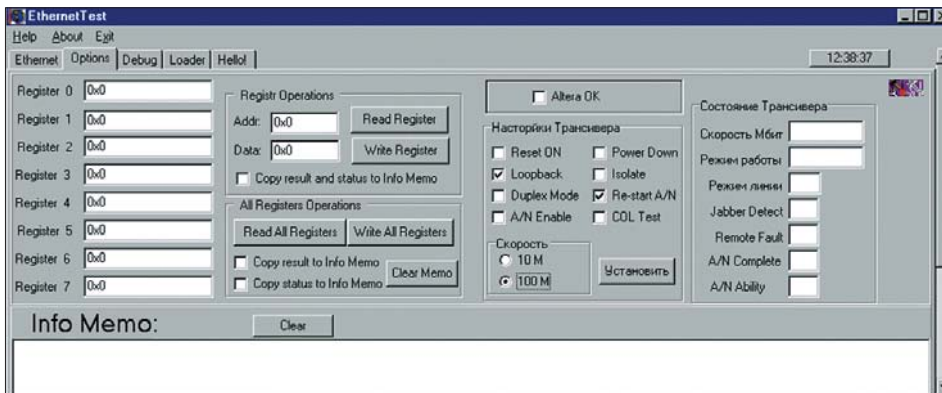


Рис. 7. Окно тестовой программы для чтения регистров трансивера и записи режимов работы

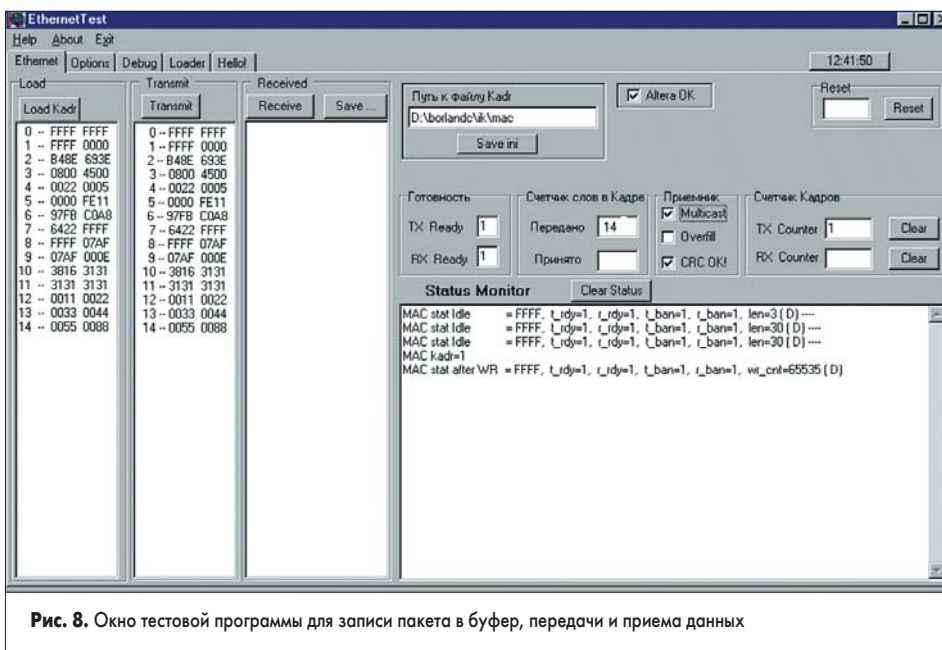


Рис. 8. Окно тестовой программы для записи пакета в буфер, передачи и приема данных

или прерывания, означающего состояние «буфер пуст наполовину». Поскольку автомат управления приемным FIFO производит блокирование чтения слов данных после получения признака «конца кадра», то прочитать «лишние» данные из следующего пакета невозможно. Эта блокировка будет снята только при чтении слова состояния по адресу регистра статуса.

2. Читается слово контроля и производится анализ битов регистра контроля. Если принятый кадр не имеет признаков ошибок при приеме (то есть, нет ошибок по CRC, не было переполнения буфера, не прихо-

дил сигнал RX_ER и т. д.), то принятый кадр считается достоверным.

3. Читается слово состояния. Производится анализ битов регистра статуса. Принимается решение о продолжении приема данных из буфера.

В том случае, если контроллер имеет каналы DMA, процессы приема и передачи значительно упрощаются, поскольку примененный в данном проекте принцип организации FIFO позволяет провести аппаратное согласование всего тракта как на прием, так и на передачу. Для передачи необходимо просто указать адрес пакета в поле памяти и его дли-

ну. По окончании передачи или при ошибке передачи будет получено прерывание. То же самое и для канала приема: необходимо только указать адрес в поле памяти, куда должен быть записан пакет. По окончании приема или при ошибке приема будет получено прерывание. Такой способ работы значительно уменьшает нагрузку на хост-процессор, но требует некоторого увеличения задействованного в FPGA ресурса.

При отладке проекта была разработана тестовая программа, позволяющая загружать FPGA, контролировать режим работы трансивера, передавать и принимать пакеты. На рис. 7 приведено окно тестовой программы, позволяющее читать регистры трансивера, производить запись режимов работы. На рис. 8 приведено окно программы, при помощи которого производится запись пакета в буфер, инициализируется передача данных и производится прием данных. Мониторинг сети проводился стандартными программами, например netXray3.

Заключение

Приведенное в данной статье описание проекта MAC-контроллера, конечно, не является полным описанием. Но, тем не менее, оно позволяет представить суть проекта, произвести оценку требуемых для данного проекта ресурсов, оценить его трудоемкость. В данной статье более подробно описаны те проблемы, которые наиболее часто поднимаются в телеконференциях, а следовательно, интересуют разработчиков.

Литература

1. А. Сигаев. Embedded Internet // «Компоненты и Технологии». 2000. № 2.
2. Н. Олифер, В. Олифер. Высокоскоростная технология Fast Ethernet (IEEE 802.3u). Центр Информационных Технологий (lxt972d.pdf).
3. Л. Куин, Р. Рассел. Fast Ethernet. BHV. 1998.
4. М. В. Кульгин. Коммутация и маршрутизация IP/IPX трафика. М.: Компьютер Пресс. 1998.
5. Implementing CRCs in Altera Devices and ALTERA CRC MegaCore Function, Parameterized CRC Generator/Checker Data Sheet. ALTERA AN049.