

**ПОВЫШЕНИЕ СКОРОСТИ ПЕРЕДАЧИ  
ДАННЫХ САН ПАКЕТА**

Нгуен Динь Туан\*

В данной работе рассмотрен вариант возможной модернизации стандартного сетевого протокола CAN. Малая длина стандартного CAN пакета ограничивает область его применения функцией передачи команд. Целью модернизации является создание возможности передачи длинных пакетов (более 512 бит) данных в уже развернутой сети между модернизированными - быстрыми узлами, с сохранением возможности работы остальных узлов в формате стандартного CAN протокола. Это, дополнительно, позволит использовать CAN для межузловой передачи первичных данных. Результатом выполненной работы является один из способов повышения скорости передачи данных CAN пакета. Это очень важно для систем, которые понадобятся передавать большие объемы данных.

## 1. Постановка задачи и способ решения

Информационное взаимодействие объектов в современных системах управления строится на основе технологии сетевой информационной среды. Для систем управления технологическими процессами, необходимо гарантировать работу всех компонентов автоматизированной системы в режиме "жесткого" реального времени. Самый существенный параметр информационной среды, влияющий на масштаб реального времени системы управления - скорость передачи пакетов данных.

## 1.1. Анализ задачи

Из-за требования совместимости скоростных узлов со стандартными необходимо:

- Сохранить в новом формате части стандартного пакета, необходимые для его восприятия и пропуска медленным узлом.
- Использовать в быстром режиме существенные черты стандартного пакета, не меняя принципов работы абонентов сети для пользователей.
- Разработать механизм передачи данных между быстрыми и медленными узлами.

Кроме того, необходимо выбрать в стандартном пакете признак, разделяющий медленные и быстрые пакеты.

По стандарту ISO11898 сеть CAN достигает максимальной скорости только до 1 Мбит/с. В работе исследуется возможность повышения скорости передачи данных в сети CAN методом использования сочетания протокола CAN, в фазе вхождения в связь, и быстрой передачи данных через приёмопередатчики стандарта RS485 во временном интервале поля данных CAN пакета.

### 1.2. Способ решения задачи

Идея заключается в использовании стандартного формата CAN пакета во всех фазах транзакции, кроме фазы передачи данных. Быстрые и медленные пакеты разделяются по адресному принципу - по старшему биту поля ID, что позволяет заранее сообщить медленному узлу о необходимости пропуска быстрого пакета. Быстрые узлы, после анализа адреса, готовы к работе в фазе передачи данных на высокой скорости. На физическом уровне, способ скоростной передачи данных не должен нарушать требований к уровням сигналов CAN сети, но может передавать лог. 1 и лог. 0 по-другому.

Мостом между медленными и быстрыми узлами может быть любой быстрый узел, т.к. стандарт CAN разрешает множественную адресацию узла. Быстрый узел в разных фазах сетевой транзакции работает на двух битовых скоростях, поэтому его легко сделать комбинированным, способным и обрабатывать сообщения в быстром и стандартном CAN формате. Далее в тексте, понятия быстрого и комбинированного узлов не различаются.

### 1.3. Список задач по реализации проекта

Для решения поставленной задачи необходимо:

- Определить состояние свободной шины.
- Выполнить процедуру арбитража - при вхождении в связь в режиме передачи и селекции адреса - в режиме приёма, с настройкой на скорость приёма, в зависимости от значения поля ID.
- Обмен данными производить с заданной полем ID скоростью.
- Остальные фазы пакета отработать на стандартной скорости.
- Самая главная задача - найти способ скоростной передачи в фазе данных.



## 2. Описание протокола CAN

### 2.1. Основные понятия сети CAN

CAN представляет собой сеть с общей средой передачи данных. Это означает, что все узлы сети одновременно принимают битовый поток, передаваемый по шине. Машины состояния всех узлов сети CAN работают синхронно - когерентно. Все узлы сети анализируют весь передаваемый трафик по шине, а CAN - контроллеры узлов предоставляют аппаратную возможность выбора сообщений для данного абонента.

Сеть CAN - сеть передачи сообщений. Для каждого сообщения в сети CAN назначается ID - идентификатор, который определяет приоритет этого сообщения. Каждый узел состоит из двух составляющих. Это собственно CAN контроллер, который обеспечивает взаимодействие с сетью и реализует её протокол, и контроллер ресурсов узла, как информационный абонент сети.

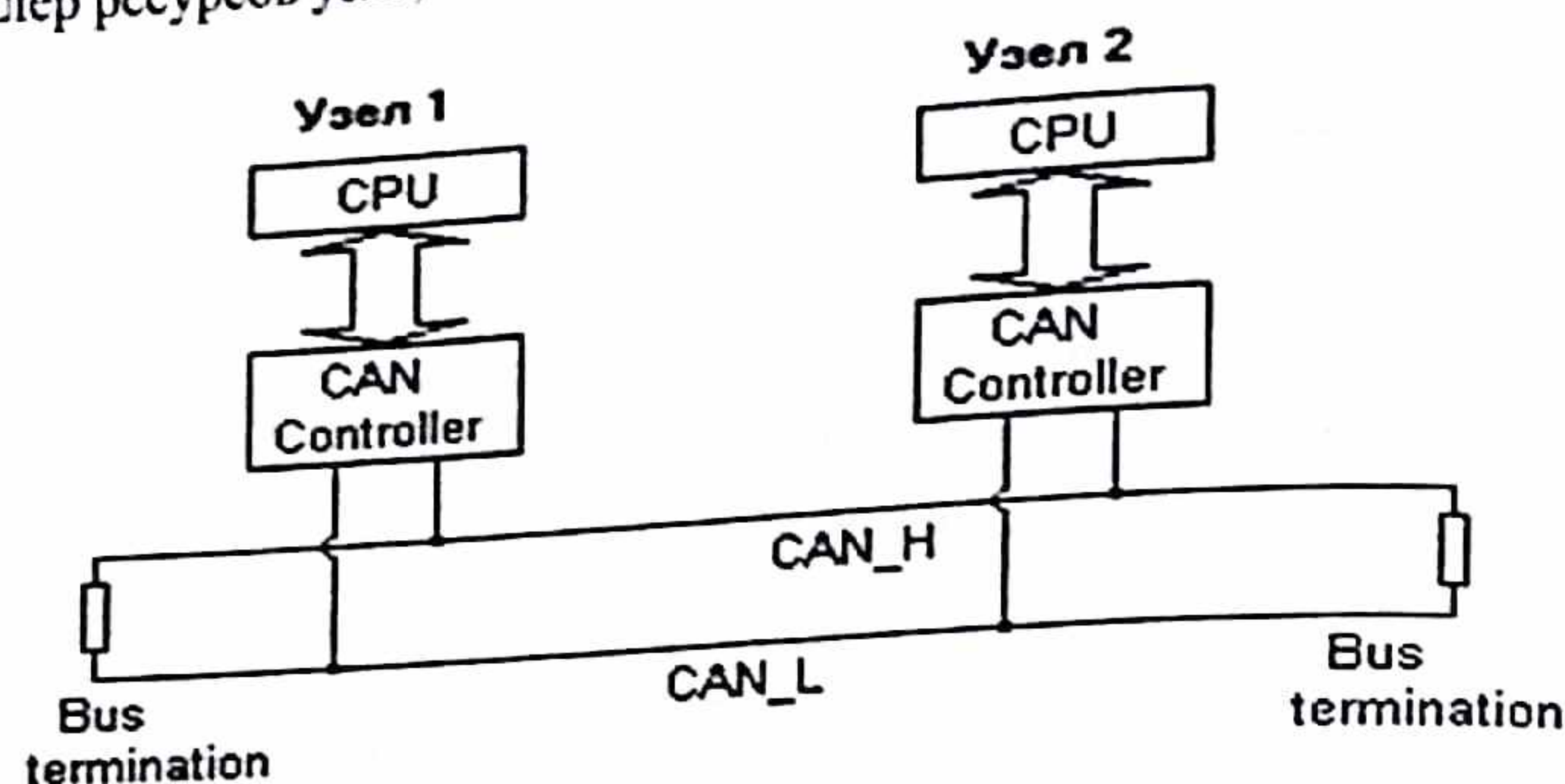


Рисунок 1. Топология сети CAN.

CAN контроллеры соединяются с помощью дифференциальной шины, которая имеет две сигнальные линии - CAN\_H (can - high) и CAN\_L (can - low). Логический ноль регистрируется, когда на линии CAN\_H сигнал выше, чем на линии CAN\_L. Логическая единица - в случае, когда сигналы CAN\_H и CAN\_L одинаковы (отличаются менее чем на 0.5 В). Использование такой дифференциальной схемы передачи делает возможным работу CAN сети в очень сложных внешних условиях. Логический ноль - называется доминантным битом, а логическая единица - рецессивным. Эти названия отражают приоритет логической единицы и нуля на шине CAN. При одновременной передаче в шину логического нуля и единицы, на шине будет сформирован и зарегистрирован узлом-приёмником только логический ноль (доминантный сигнал), а логическая единица будет подавлена (рецессивный сигнал).

### 2.2. Форматы стандартного и быстрого CAN - пакетов

#### 2.2.1. CAN - пакет стандартного формата: CAN-LS (low speed)

Такой тип пакета передается только со скоростью до 1 Мбод. Предположим, что для передачи мы используем стандартный CAN - data frame, и в поле DATA содержится 0 ÷ 7 bytes данных. Тогда биты заполняются в полях соответственно:

SOF	IDENT	RTR	IDE	R <sub>0</sub>	DLC	DATA	CRC		ACK		EOF
"0"	11 bits	"0"	"0"	"0"	"0xxx"	0-7 bytes	15 bits	"1"	"1"	"1"	7 bits "1"

Рисунок 2. Формат CAN frame low - speed.

Поскольку, решение задачи основано на доработке пакета "CAN - data frame" с полем идентификатора из 11 битов, мы можем не исследовать поля RTR.

Чтобы различать стандартный и быстрый CAN - пакеты, будем:

- Для стандартного пакета 1-ый бит в поле IDENT задавать равным "0".
- Для быстрого пакета - 1-ый бит в поле IDENT задавать равным "1".

В режиме "low - speed" процессы передачи и приёма работают в сетке сигнала "Clk1MHz", который получается из глобального синхронного сигнала "Global Clock Clk80MHz".

#### 2.2.2. CAN - пакет быстрого формата: CAN-HS (high speed)

Данные в таком пакете могут передаваться со скоростью до 10 Мбод.

В отличие от стандартного пакета, CAN - пакет быстрого формата имеет поле данных максимальной длины. Это значит, что время передачи поля данных равно времени передачи 8 байт данных на стандартной скорости, а значение поля DLC в формате High - Speed всегда равно «1000»:

SOF	IDEN	RTR	IDE	R <sub>0</sub>	DLC	DATA	CRC		ACK		EOF
"0"	11 bits	"0"	"0"	"0"	"1000"	80 bytes	15 bits	"1"	"1"	"1"	7 bits "1"

Рисунок 3. Формат CAN frame high - speed.

Первый бит в поле IDENT равен "1".

В режиме «high speed», данные в поле данных передаются и принимаются в сетке сигнала "Clk10MHz", а остальные поля - со скоростью 1MHz. Сигналы Clk1MHz и Clk10MHz получаются из глобального синхронного сигнала "Global Clock Clk80MHz".



Таким образом, поле данных (data field) в режиме «high speed» содержится  $8 \times 10 = 80$  байт данных. Это значит, что в поле данных вместо 8 байт мы можем передавать 80 байт с высокой скоростью - 10 МБод.

### 2.3. Почему поставлена задача модернизации CAN?

**Проверенный, надёжный стандарт.** Протокол CAN активно используется уже более 18 лет. Сейчас на рынке представлено много различных CAN изделий и инструментальных средств для разработки приложений на его основе.

**Аппаратная поддержка протокола.** Минимальная аппаратная реализация CAN поддерживает все фазы CAN транзакции, что позволяет быстро и легко строить высокоэффективные системы в терминах адресации типов данных, не касаясь способа их передачи.

**Различная среда передачи.** Основная среда передачи - витая пара. CAN система может также работать только на одном проводе.

**Превосходная обработка ошибок.** Протокол CAN имеет механизм, позволяющий отключать дефектный узел и не допускать блокирование сети.

**Хорошая поддержка систем реального времени.** Использование глобальных часов и широковещательный способ передачи сообщений позволяет создавать полностью синхронные системы.

**Хорошая поддержка систем, управляемых событиями.** Так как все CAN узлы слушают все сообщения, просто реализуются приложения, работающие по событийному управлению.

**Ориентирование на распределенные системы управления.** Протокол CAN очень хорошо подходит для построения распределенных систем управления. Используемый метод арбитража для определения приоритета сообщения и широковещательная передача позволяют просто и оптимально проектировать такие системы.

## 3. Реализация комбинированного CAN узла

### 3.1. Построение приёмопередатчика комбинированного узла

Обычные, CAN приёмопередатчики не могут работать на большой скорости, но уровни сигналов линиях CAN сети совпадают с уровнями сигналов интерфейса RS485, а скорость передачи интерфейсом RS485 существенно выше. Поэтому, предлагается использовать приёмопередатчик RS485 для реализации физического уровня быстрого узла CAN сети в фазе пересылки данных.

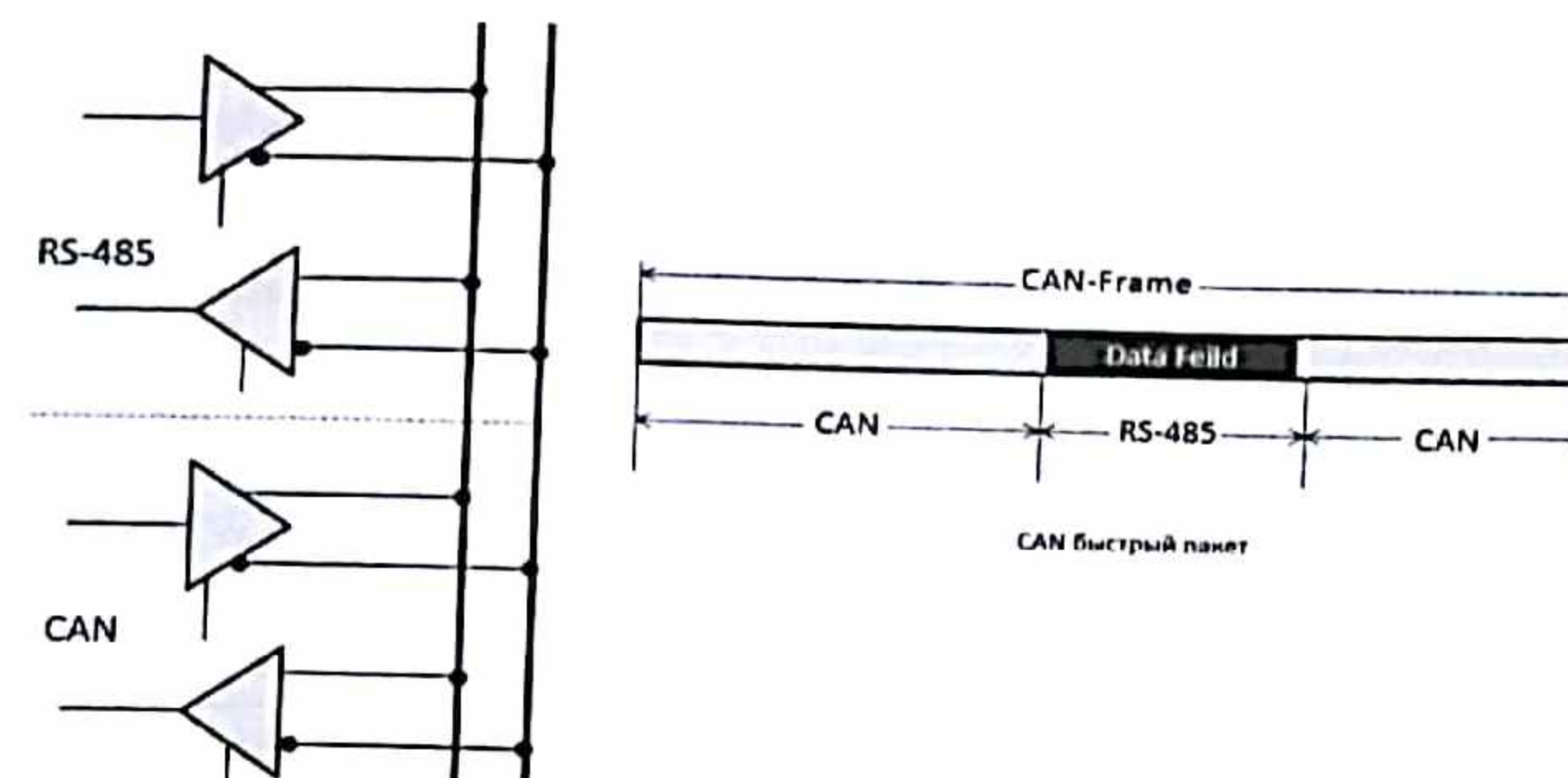


Рисунок 4. Физический уровень комбинированного CAN узла.

Как мы уже предположили выше, большинство изменений будет происходить в фазе пересылки данных. В этой фазе, для быстрой пересылки мы будем использовать приёмопередатчик RS-485 на скорости 10 МБод. Обмен другими полями и для быстрого и для медленного пакета будет происходить через стандартный CAN приёмопередатчик.

### 3.2. Реализация комбинированного CAN узла

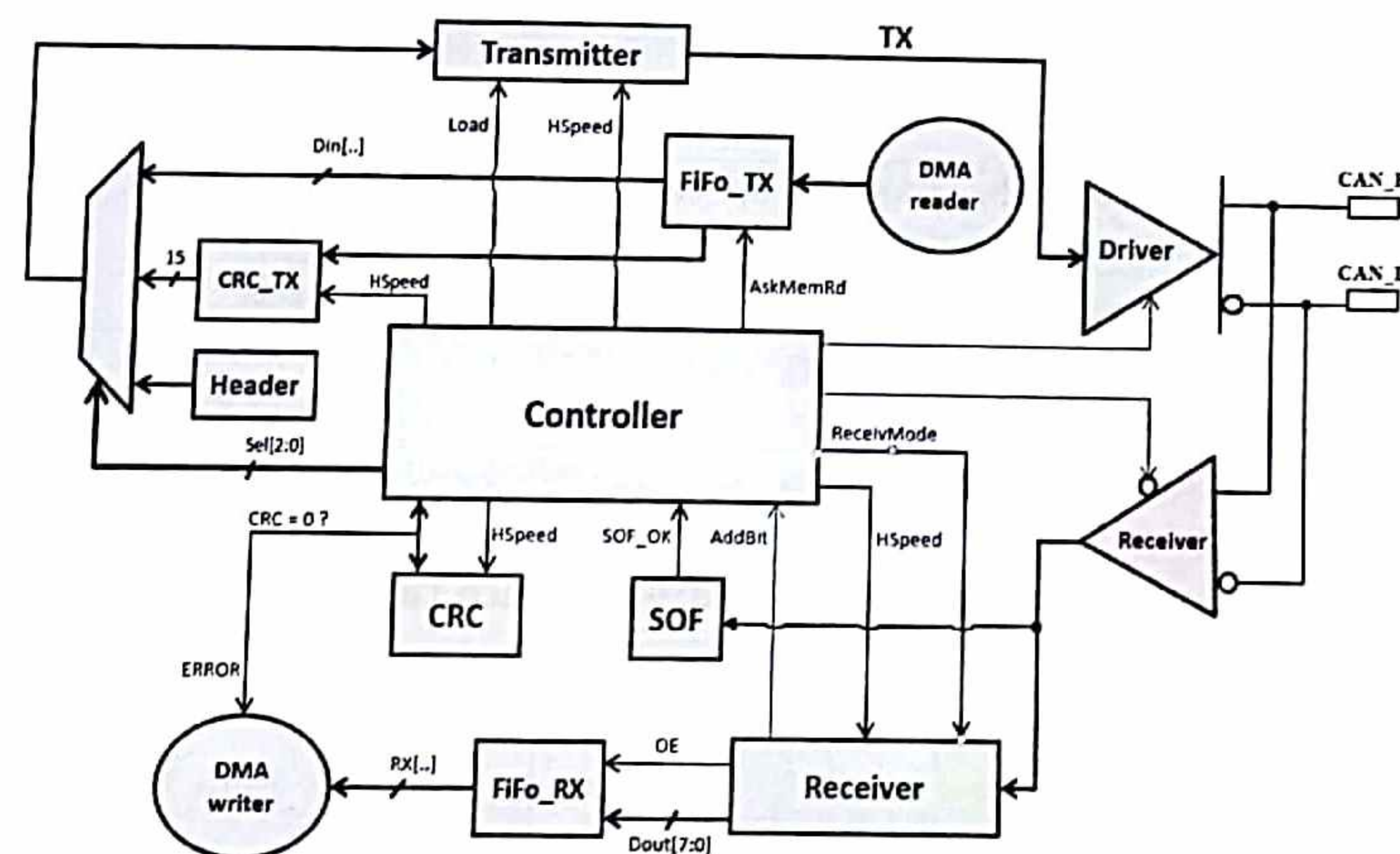


Рисунок 5. Структурная схема комбинированного CAN модуля.



Рассмотрим отдельно тракты передачи и приёма data-пакета:

### 3.2.1. Список задач, решаемых при передаче пакета

Таблица 1

Функция	Модуль
Формирование пакетов, которые должны передаваться по шине в буфере передатчика	FIFO_TX
Определение момента освобождения шины, для запуска передачи пакета	SOF
Состязание за доступ	Arbitr
Передача пакета в линию, при благоприятном арбитраже с определённой заданной скоростью	P2S (Parallel to serial), CRC_TX

### 3.2.2. Список задач, решаемых при передаче пакета

Таблица 2

Функция	Модуль
Битовая синхронизация с другими узлами сети (формирование локального сигнала - разделителя битов пакета)	SOF
Определение начала пакета данных, обнуление контрольной суммы	SOF, CRC
Селекция пакета	Arbitr
Приём слов пакета, формирование контрольной суммы	S2P (Serial to parallel), CRC
Анализ контрольной суммы	RX_control
Формирование сигналов ошибки приёма или готовности пакета	RX_control
Приёмный буфер, логика управления очередью	FIFO_RX

## 4. Проверка работы узлов с пакетами разного типа на модели

При нормальной работе CAN, машины состояний передатчика и всех остальных узлов сети переключаются побитно-синхронно.

Во времени работы CAN - приёмопередатчика все узлы, которые подключились к шине, должны работать синхронно и когерентно.

Рассмотрим некоторые случаи взаимодействия разнотипных узлов:

**4.1. Первый случай:** Передатчик передает медленный пакет, приёмник принимает этот пакет с низкой скоростью. Проверяется работа модулей при передаче и приёме на стандартной скорости.

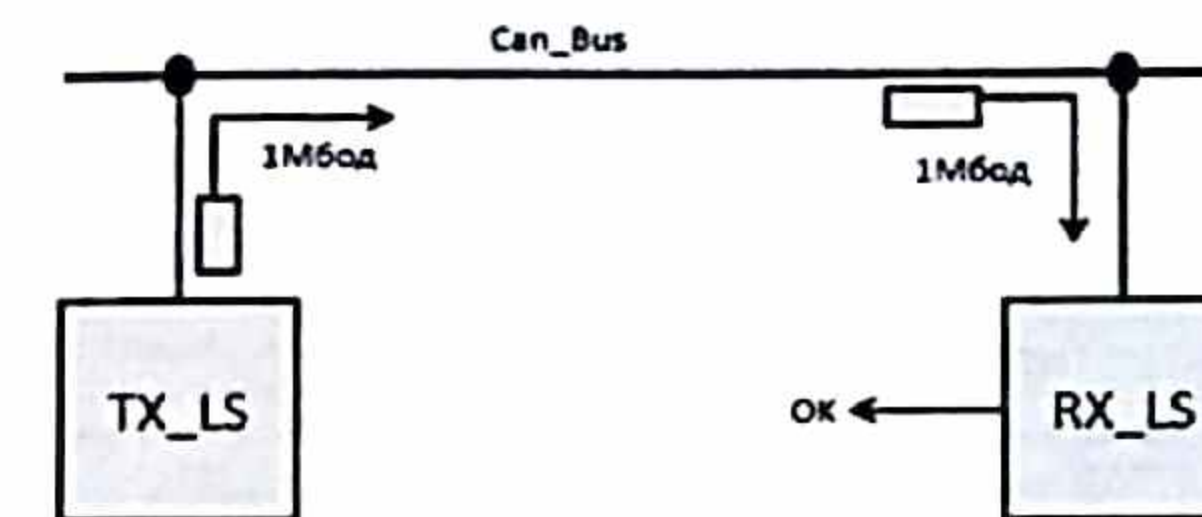


Рисунок 6. Приём медленным узлом стандартного пакета.

В этом случае для передачи CAN пакет с данными длины до поля данных:

SOF	IDENT	RTR	IDE	R <sub>0</sub>	DLC	DATA
0	00010001011	0	0	0	0001	10111001

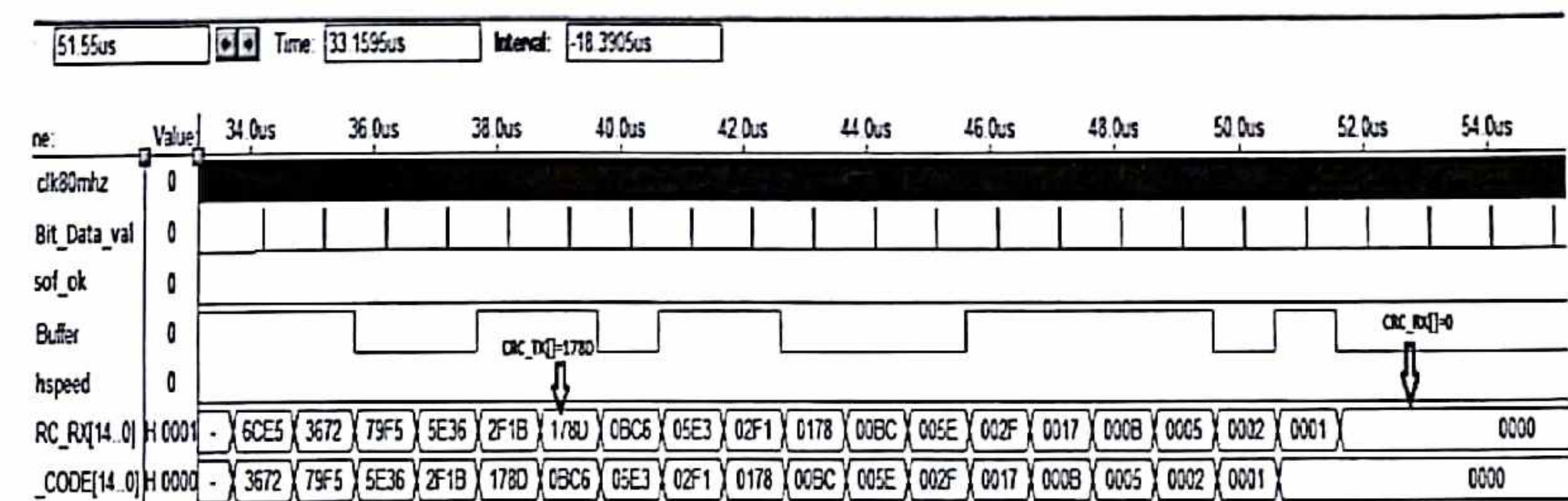


Рисунок 7. CRC\_RX в случае LS – LS.

Получаем:

CRC\_TX [...] = 178D = 001 0111 1000 1101

CRC\_RX [...] = 0. Это значит, что узел правильно принял пакет.

**4.2. Второй случай:** Передатчик передает медленный пакет, приёмник принимает этот пакет с высокой скоростью. Приёмник должен зафиксировать пакет с чужим адресом и возможную ошибку контрольного кода.

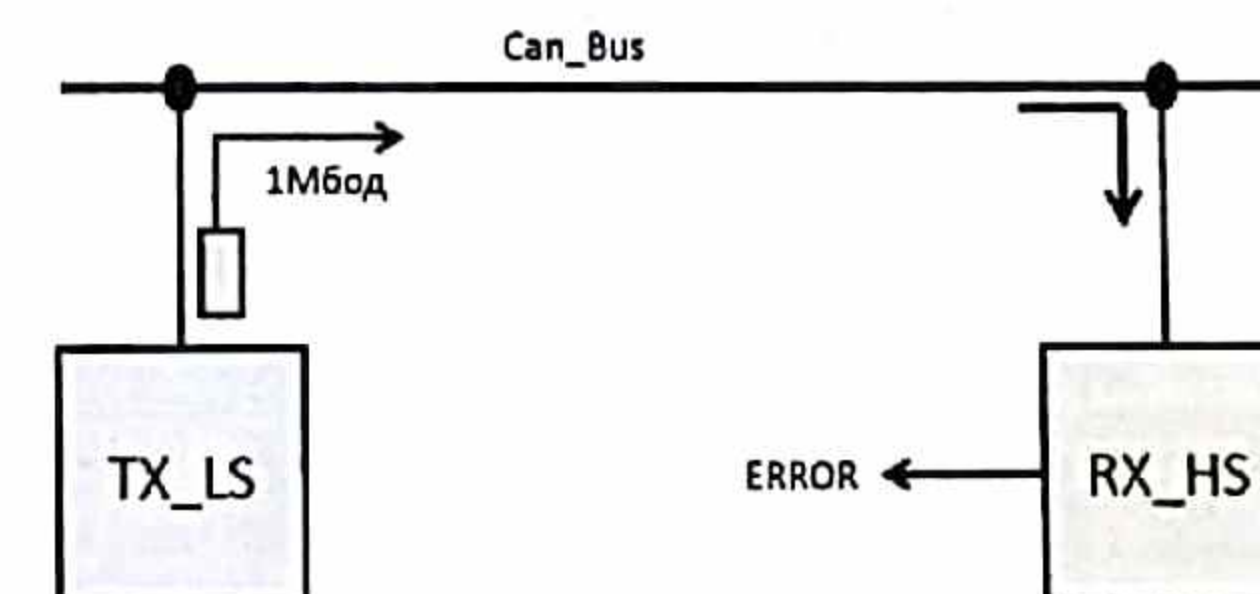


Рисунок 8. Приём быстрым узлом стандартного пакета.



Берём такой пакет как в первом случае

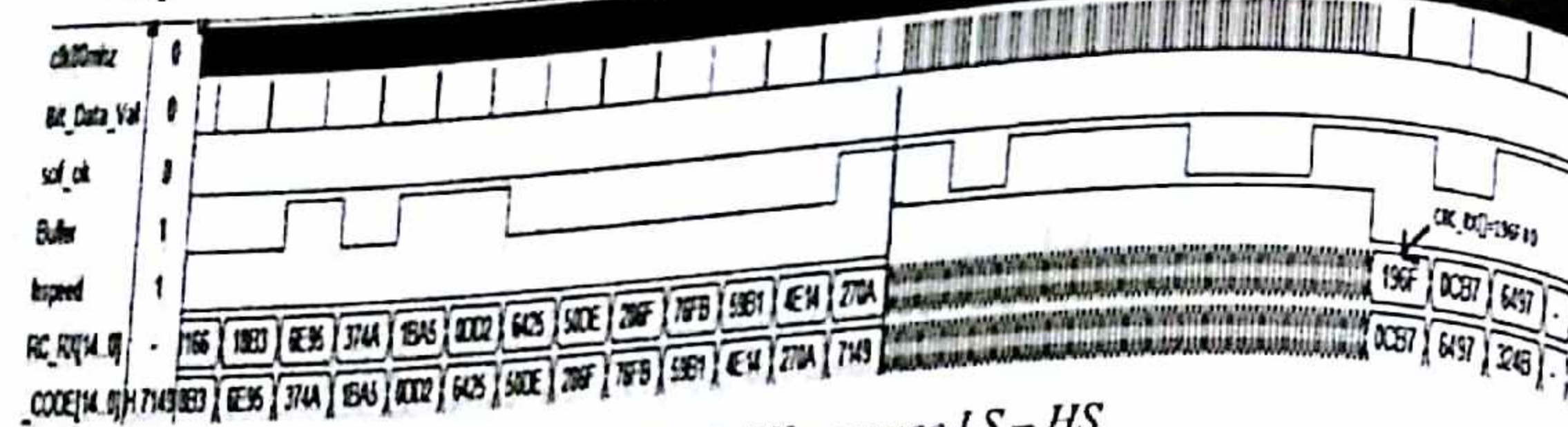


Рисунок 9. CRC\_RX в случае LS-HS.

Получаем:

Получаем:

$$\text{CRC\_TX} [...] = 178D = 001\ 0111\ 1000\ 1101$$

CRC\_RX [...] = 196F не равен 0. Это значит, что приёмник с высокой вероятностью получил неправильную.

**4.3. Третий случай:** Передатчик передает быстрый пакет, и приёмник принимает этот пакет с высокой скоростью. Приём данных быстрым узлом должен пройти без ошибок.

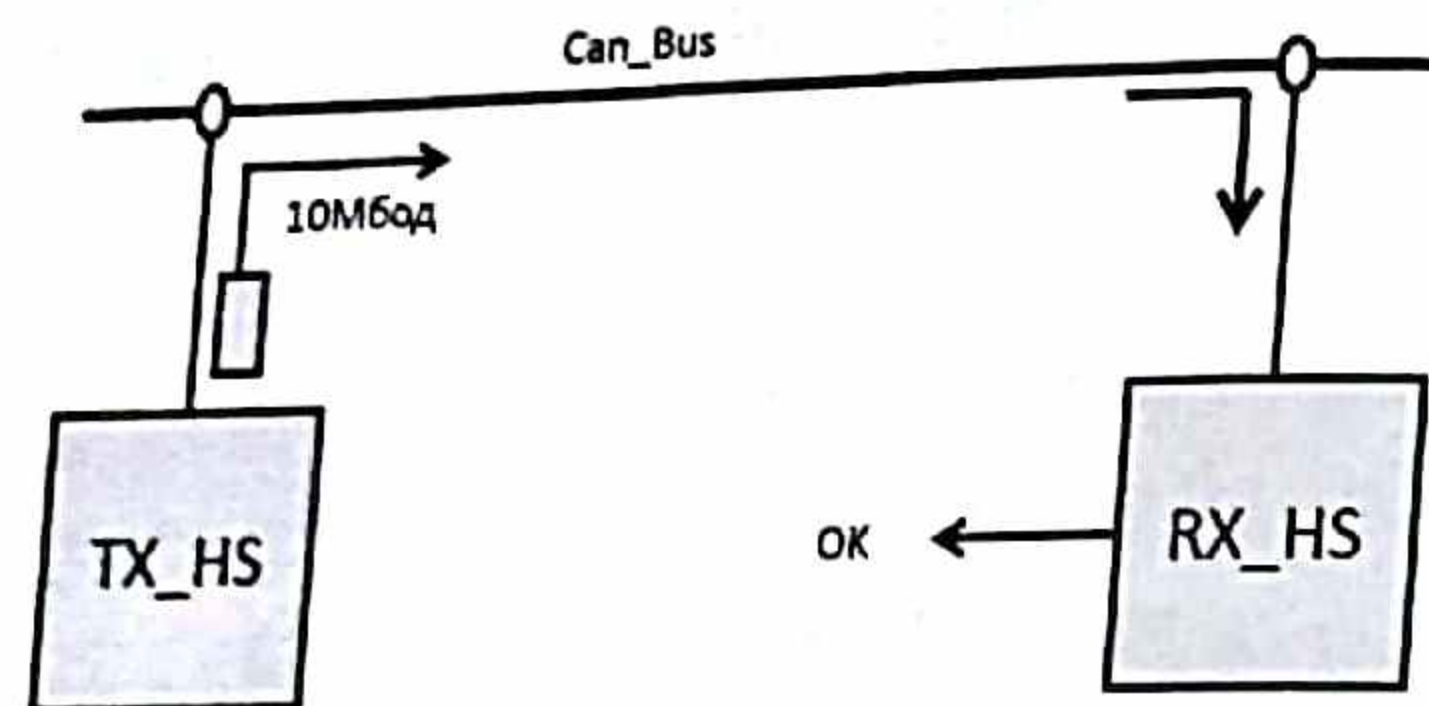


Рисунок 10. Приём быстрым узлом быстрого пакета.

В этом случае для передачи CAN пакет с данными длины до поля данных:

SOF	IDENT	RTR	IDE	R <sub>0</sub>	DLC	DATA
0	10000001011	0	0	0	1000	111...111 (8 bytes "I")

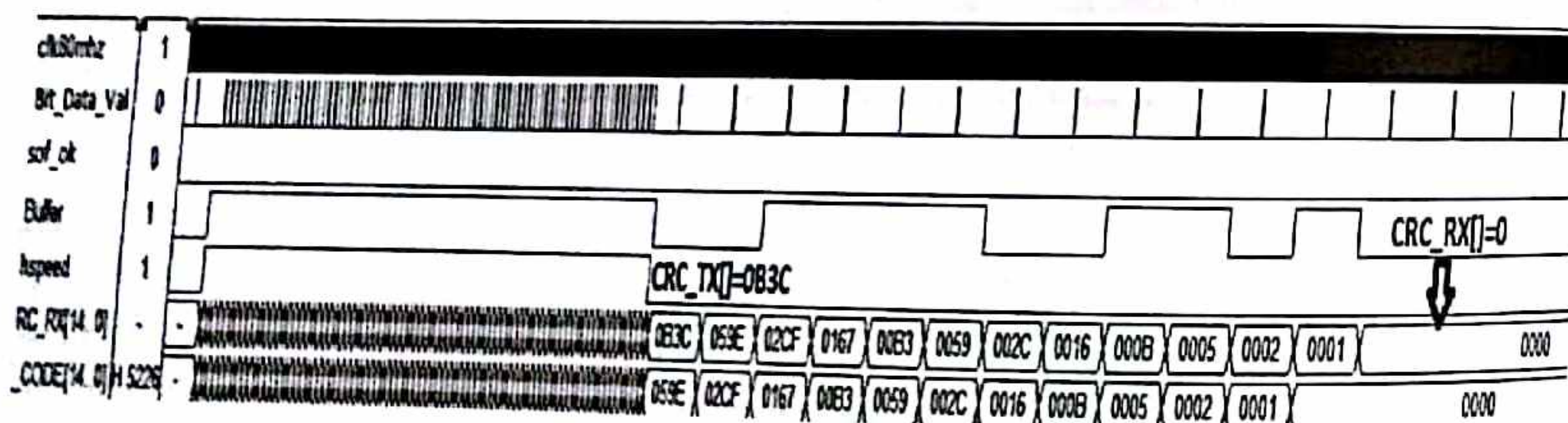


Рисунок 11. CRC\_RX в случае HS-HS.

Получаем:

CRC\_TX[...] = 0B3C = 000 1010 0011 1011

CRC\_RX [...] = 0. Это значит, что узел правильно принял пакет.

**4.4. Четвертый случай:** Передатчик передает быстрый пакет, и приёмник принимает этот пакет с низкой скоростью. Приёмник должен зафиксировать пакет с чужим адресом и возможную ошибку контрольного кода.

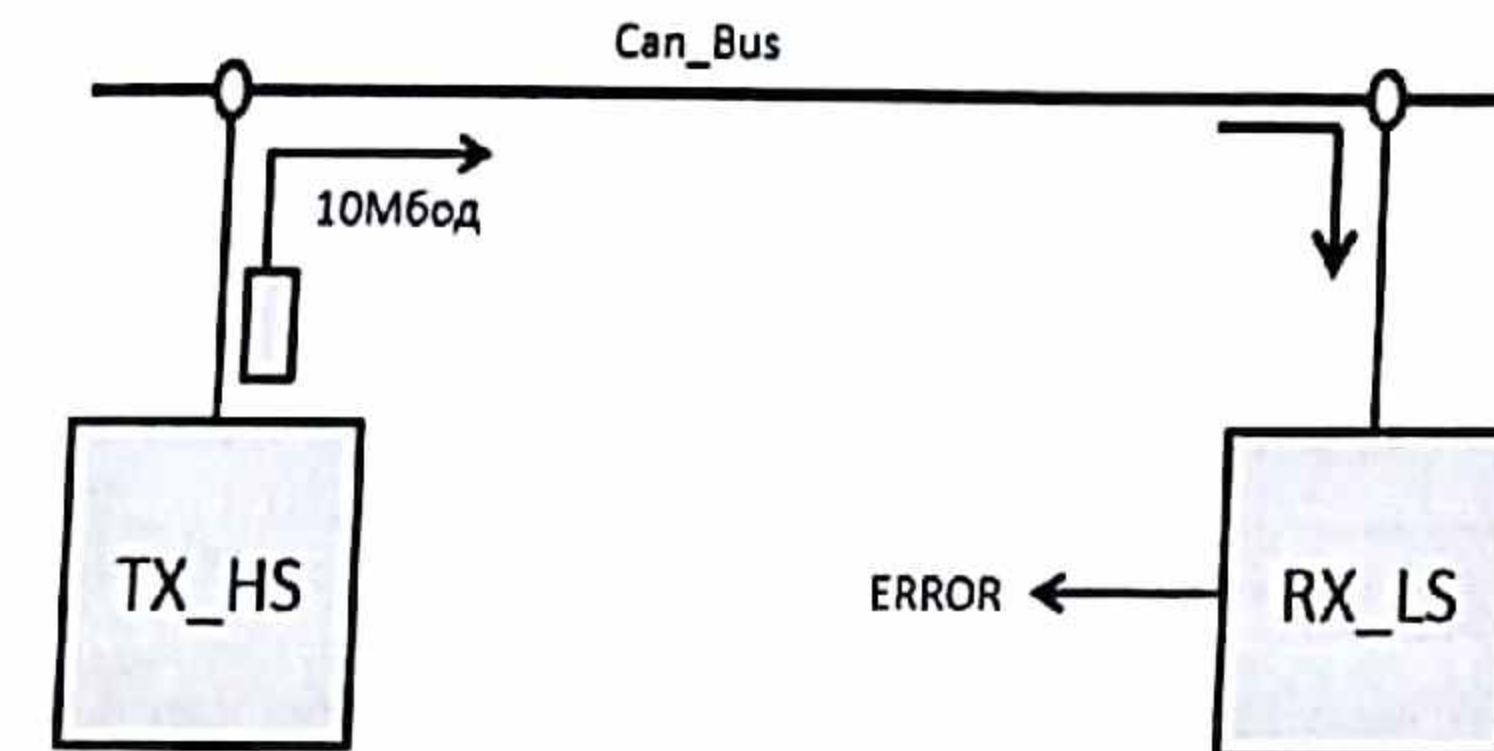


Рисунок 12. Приём быстрым узлом быстрого пакета.

Берём такой пакет как в третьем случае:

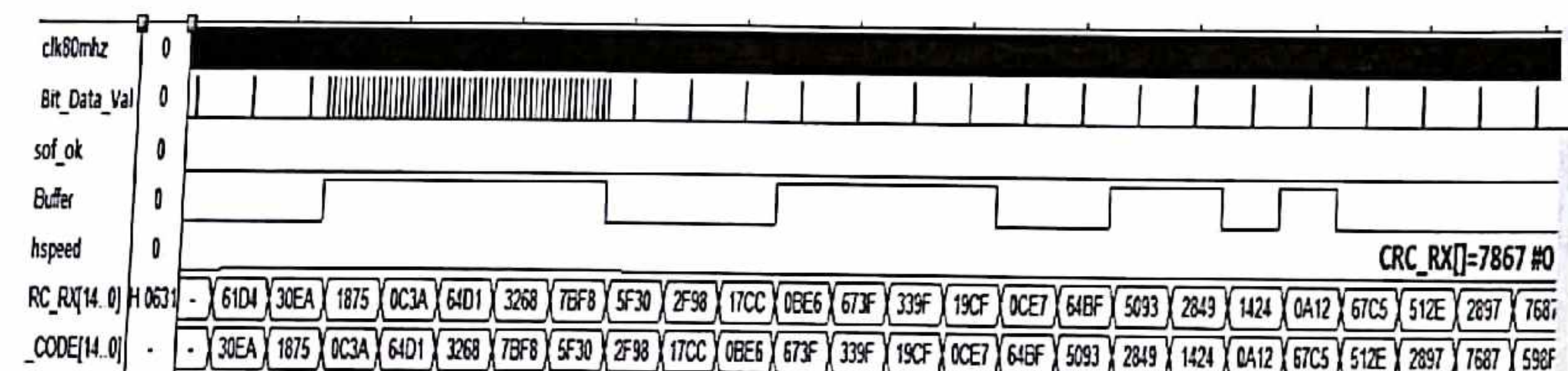


Рисунок 13. CRC\_RX в случае HS – LS.

Получаем:

CRC\_TX [...] = 0B3C = 000 1010 0011 1011

CRC\_RX [...] = 7687 не равен 0, это значит, приёмник с высокой скоростью принимает медленный пакет неправильно.

## 5. Заключение

Моделирование подтвердило правильность работы комбинированных узлов в CAN сети. Взаимодействия между двумя быстрыми и так же между двумя медленными узлами происходят без ошибок. Взаимодействие между быстрым и медленным узлами корректно завершается игнорированием приёма данных от разнотипного узла.



### **Недостатки:**

- Максимальная длина сети обратно пропорциональна скорости передачи.
- Большой размер служебных данных в пакете (по отношению к полезным данным).
- Отсутствие единого общепринятого стандарта на протокол высокого уровня, однако же, это и достоинство. Стандарт сети предоставляет широкие возможности для практически безошибочной передачи данных между узлами, оставляя разработчику возможность вложить в этот стандарт всё, что туда сможет поместиться.

### **Список литературы**

1. Антонов А.П. *Язык описания цифровых устройств. AlteraHDL*, 2001.
2. Robert Bosch GmbH. *CAN Specification 2.0*. Sep. 1991.
3. *CAN интерфейс (Control Area Network)*. <http://www.ahdl-altera.narod.ru/ru/can.htm>.
4. *RS485 serial information*. <http://www.lammertbies.nl/comm/info/RS-485.html>.
5. Tạ Đức Anh. *Controller Area Network*.