

МОДУЛ ЗА РЕАЛИЗАЦИЯ НА “LIN” ИНТЕРФЕЙС

Благомир Росенов Дончев, Красимир Христов Христов

Технически Университет – София, ECAD лаборатория

donchev@ecad.vmei.acad.bg, khhristov@yahoo.com

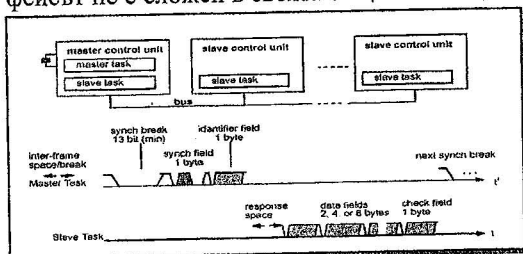
Donchev B.R., Hristov K.H., LIN protocol module. The goal of this design is to present an implementation of LIN protocol in programmable logic areas. Top-down approach has been used during the design stage and VHDL has been applied for the design description. The presented module meets the requirements of LIN Protocol Specification, Revision 1.2, which is the latest up to now. LIN interface is created to be used in cases when high speed and high safety of transmitted data are not critical. Directed entirely in automotive sector applications it uses a single 12V wire to transmit data. Its main properties are: low cost silicon implementation, single master/multiple slave configuration, self-synchronization without quartz or ceramics resonator in the slave nodes and speed up to 20 kbit/s.

1. ВЪВЕДЕНИЕ

През последните десетина-двадесет години електрониката навлиза все по-дълбоко в съвременната автомобилна индустрия. С прецизността си, възможността да се създават надеждни електронни прибори и лесният контрол над тези прибори електрониката недвусмислено доказва ефективността си. Но с “поставянето” на електронни прибори на борда на един автомобил се появи и необходимостта да се осигури лесен, сигурен и евтин начин за комуникация по между им, т.е. появяват се интерфейсите за връзка между електронните компоненти.

Разбира се в търсене на оптималното решение – високо качество, ниска себестойност, голяма бързина и малка консумация се появяват различни интерфейси, които се обособяват като най-подходящи за определено приложение. В случаите когато не са необходими голяма сигурност на информацията и високи скорости на обмен, един от подходящите варианти е LIN. LIN интерфейсът не е сложен в своята същност. В идеята за създаването му от LIN консорциумът е залегнала основната цел да се изгради нова,

евтина серийна комуникационна система с намерение да бъде въведена в използваните електронни устройства в превозните средства и да допълни съществуващите автомобилни интерфейси. LIN реализира ефективна комуникация с ин-



Фиг.1 Структура на съобщението

телигентни сензори и актюатори, където капацитета и гъвкавостта на CAN не

са необходими. Спецификацията е съобразена с ISO 9141 стандарта. Комуникацията е базирана на SCI (UART) информационният формат, а концепцията е единичен главен (управляващ) възел – MaCU (Master Control Unit) и множество подчинени възли SCU (Slave Control Unit) – *фиг. 1*. На практика броят на SCU не надвишава дванадесет, което е резултат от не големия брой на дефинираните валидни стойности на идентификаторите (64) и относително ниската скорост на обмен на информация. Информацията се разпространява по единична 12 волтова линия (колкото е захранващото напрежение в един автомобил), а синхронизацията на възлите се постига без да е необходим повече от един стабилизирани източник на тактов сигнал, т.е. чрез керамичен или кварцов резонатор свързан към MaCU (SCU възлите могат успешно да работят с не толкова прецизни генератори на тактови сигнали). Всеки възел, участник в комуникацията на една LIN мрежа не се “интересува” как е конфигурирана системата, освен от това дали има функциониращ MaCU възел. По този начин се дава възможност да се добавят и премахват възли без да са необходими някакви хардуерни или софтуерни промени в остиналите участници.

Информацията по LIN мрежата се разпространява в строго определен формат на съобщението с променлива дължина. Всяко съобщение може да съдържа два, четири или осем информационни байта плюс още три за контрол и защита на информацията. Започва със синхронизиращо прекъсване (Synch Break – *фиг. 1*), поле за самосинхронизация (Synch Field) и поле на идентификатора (Identifier Field). Всички те се изпращат от т.нар. главна задача (Master Task). В нейн отговор се явява подчинената задача (Slave Task), която пренася информацията по мрежата. MaCU може да предава както Master Task, така и Slave Task, докато SCU – само Slave Task.

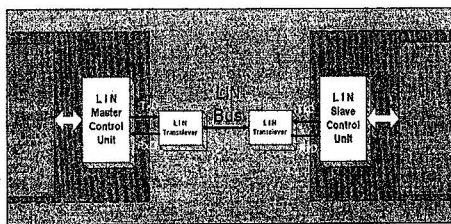
За да се намали консумацията в една LIN мрежа съществува възможност всеки възел да бъде поставен в Sleep mode режим, т.е. без вътрешна активност и с пасивен драйвър. Единствено MaCU има правото да изпраща команда, която се приема от всички подчинени възли, и която им указва, че системата преминава в Sleep mode.

LIN интерфейсът не дава възможност за директно сигнализиране при откриване на грешка. Ако MaCU отчете грешка (например липсваща Slave task или грешка при сверяването с Checksum полето) може да промени последователността на предаване на съобщенията, а ако SCU отчете грешка, то тя ще бъде запомнена като диагностична информация и предоставена на MaCU при поискване. Диагностичната информация може да бъде предадена чрез информационните полета на едно съобщение. Съществуват шест вида грешки.

2. ПРИНЦИП НА ДЕЙСТВИЕ

2.1 ВЪТРЕШНИ БЛОКОВЕ

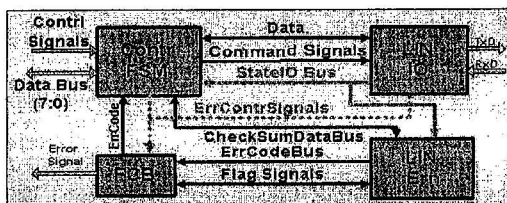
Тъй като LIN модулът е разработен с помощта на език за хардуерно проектиране от високо ниво, се постига сравнителна независимост от средата



Фиг.2 Блокова схема на свързване

и с микроконтролер или друга интегрална схема. Възможно е и вграждането на модулите в бъдещи проекти за разработка на интелигентни сензори, драйвери или микроконтролери със собствени вътрешни блокове за комуникация чрез LIN стандарта. Работната честота е 8MHz.

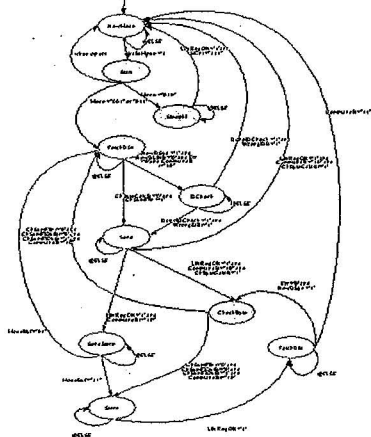
Основните вътрешни архитектури на главния и подчинения възел са еднакви (фиг.3), като във всеки от двата случая е необходимо дизайнът да изпълнява и някои допълнителни функции. Основни изграждащи интерфейса подмодули са: ContrFSM, LIN_IO, FCB, LIN_Err.



Фиг.3 Вътрешна структура

ContrFSM

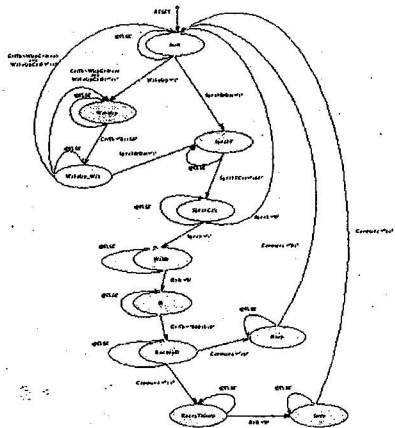
Блокът е хардуерен модел на краен автомат описващ отделните състояния, които LIN интерфейсът може да заема по време на работата си. Започва с инициализация на режима на работа и проверка на ID полето (MaCu – фиг.4) или с детектиране на ново съобщение (SCU). Преминва се към приемане на данни от LIN мрежата и извеждането им на DataBus шината или обратното. Успоредно с това се управлява и CheckSum процедурата в LIN_Err. При MaCu се осъществява и командата за SleepMode.



Фиг.4 Блокова схема на крайният автомат на ContrFSM MaCu

LIN_IO

Блокът служи за реализация на интерфейса с LIN мрежата. Отново чрез краен автомат се реализира изпращането или приемането на Master task и Slave task. В SCU варианта (фиг.5) е реализиран и алгоритъм за WakeUp процедура и проце-



Фиг.5 Блокова схема на крайният асломат на LIN_IO_SCU

дура за самосинхронизация

LIN_Err

Предназначението му е да разпознава ситуациите, в които има възникнала грешка. Чрез таймерна система се следи за спазване на времената за приемане и предаване на съобщението. Кодира се типа на грешката

FCB

Той е в пряка връзка с блока LIN_Err, целта му е да изготвя реакцията на устройството в зависимост от възникналата грешка според описаните в спецификацията процедури. В MaCu варианта са вградени и броячи следящи поведението на всеки SCU възел.

2.2 MASTER CONTROL UNIT

В Таблица 1 са описани входно-изходните сигнали на блока. Приемането и предаването на данни се осъществява по осем битова шина, по която се зарежда и вътрешен контролен регистър. Скоростта на предаване се определя от битове (7:6)BaudR, режима на работа от (5:4)Mode, а номера на отговарящото SCU от (3:0)SCU_N. Комбинация от "10" на NewDSys оказва наличието на валидни данни на DataBus шината за контролния регистър, а "11" – данни за предаване Комбинации "00" и "01" са невалидни. При прочитане на подадените

Сигнал	Описание
Bit r	TxD вход за следене на предаваните битове
Clk	Вход на главният системен тактов сигнал
Rst	Вход за сигнала за начално установяване
NewDSys	Вход оказващ наличието на постъпващи данни и типа им
DataBus	Информационна входно-изходна шина
RxD	Вход за данни от LIN мрежата
ErrPin	Изход за сигнализация при грешка
NewDLIN	Изход оказващ наличието на данни за предаване
TxD	Изход за данни към LIN мрежата
WakeUpSgnl	Изход сигнализиращ приета команда за WakeUp
WrongID	Изход за сигнализация за приет грешен идентификатор

Таблица 1 Описание на входно-изходните сигнали на MaCU

данни, NewDLIN се установява в логическа единица. В случаите когато се изведат данни на DataBus първоначално NewDLIN приема логическа единица, а NewDSys отговаря с "11". Блока разпознава подаденото му ID поле и сигнала

лизира (WrongID = '1'), ако то е невалидно. При приемане на WakeUp сигнал в режим SleepMode, WakeUpSgnl приема логическа единица.

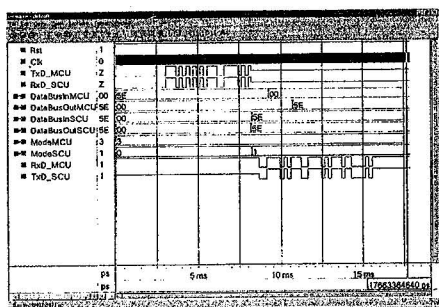
2.3 SLAVE CONTROL UNIT

Входно-изходните сигнали са идентични с MaCU. Описани са в Таблица 2. Скоростта на предаване се определя от битове (7:6)BaudR на контролния регистър, режимът на работа от (5:4)Mode, а битове (3:0) са свободни. Входовете NewDSys и NewDLIN работят както в MaCU. WakeUpSgnl = '1' сигнализира за приет WakeUp сигнал в режим SleepMode, а WakeUpFail = '1' за грешка в WakeUp процедурата.

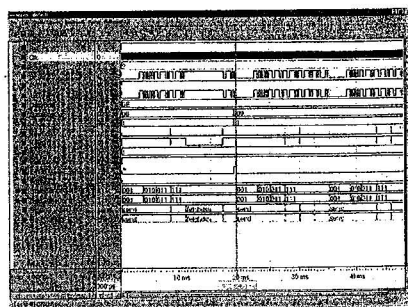
Сигнал	Описание
Bit r	TxD вход за следене на предаваните битове
Clk	Вход на главният системен тактов сигнал
DataBus	Информационна входно-изходна шина
NewDSys	Вход оказващ наличието на постъпващи данни и типа им
Rst	Вход за сигнала за начално установяване
RxD	Вход за данни от LIN мрежата
ErrPin	Изход за сигнализация при грешка
NewDLIN	Изход оказващ наличието на данни за предаване
TxD	Изход за данни към LIN мрежата
WakeUpFail	Изход сигнализиращ грешка в WakeUp процедурата
WakeUpSgnl	Изход сигнализиращ приета команда за WakeUp

Таблица 2 Описание на входно-изходните сигнали на SCU

3. СИМУЛАЦИИ И СИНТЕЗ



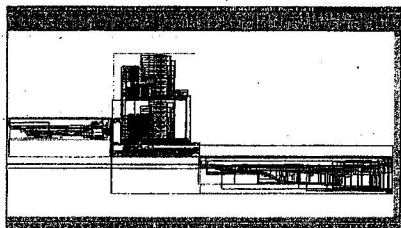
Фиг.6 Предаване на данни от SCU към MaCU



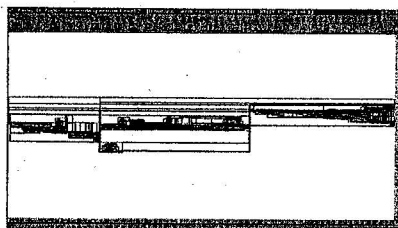
Фиг.7 Изпращане на данни от MaCU към SCU с Slave-Not-Responding-Err

На фиг.6 са показани симулационни резултати при предаване на данни от подчинения към главния възел и обратното – фиг.7. Сигналите на информационните шини на двата възела са взети преди обединяващите буфери. Представена е и работата на крайните автомати в ContrFSM и LIN_IO.

Като резултати след синтез са включени фиг.8 и фиг.9.



Фиг.8 Синтезирана схема на MACU блока



Фиг.9 Синтезирана схема на SCU блока

4. ЗАКЛЮЧЕНИЕ

Разработен на базата на един доста нов стандарт (създаден 1999г.) представеният дизайн предлага хардуерен модел на устройство за реализация на LIN интерфейс. За целта е използван език за описание от високо ниво VHDL и програмируема логика от тип FPGA на фирмата Xilinx. Основните причини за избора са огромните улеснения в процеса на проектиране, голямата гъвкавост на дизайна, кратките срокове за реализация и широките възможности за по-нататъшно развитие.

За основен източник на информация бе използвана версия 1.2 на LIN спецификацията. В процеса на работа бе направено обстойно проучване на излезлите до момента на пазара LIN продукти на фирми като : Mitsubishi, Philips, ST, Fujitsu, Motorola, PIC, Atmel, Seimens, Infenion, Melexis и NEC, което допълни общите насоки и изисквания към дизайна.

Проектираният модул бе разработен в EСAD лаборатория към Техническият университет. Той може да бъде реализиран като самостоятелна интегрална схема или безпроблемно да вземе участие като допълнителен блок в бъдещи по-големи проекти.

5. ИЗПОЛЗВАНА ЛИТЕРАТУРА

1. LIN Specification, Revision 1.2, Nov 17, 2002;
2. LIN (LOCAL INTERCONNECT NETWORK) SOLUTIONS – AN1278, Application Note, 2002 STMicroelectronics;
3. Local Interconnect Network (LIN) Demonstration – Motorola Semiconductor, Application Note – AN2103, January 2002;
4. Speks J.Will, A. Rajnak, LIN – protocol, development tools, and software inefaces for local interconnect network in vehicles;
5. Christopher A.Lupini, Mutiplex bus progression Delphi Delco Electronics System;
6. Roth, H. Charles Digital system design using VHDL , PWS Publishing Copmany, 1998;
7. Philips smart power solitions, Philips Semiconductors, February 2002;