

# CAST



## LIN Controller Core

The LIN core is a communication controller that transmits and receives complete LIN frames to perform serial communication according to the LIN Protocol Specification 2.1. It uses a single master/multiple slave concept for message transfer between nodes of the LIN network. The LIN can be implemented as a master or as a slave. The message transfer can be controlled via a micro controller interface and a LIN transceiver is needed for the connection to the LIN bus.

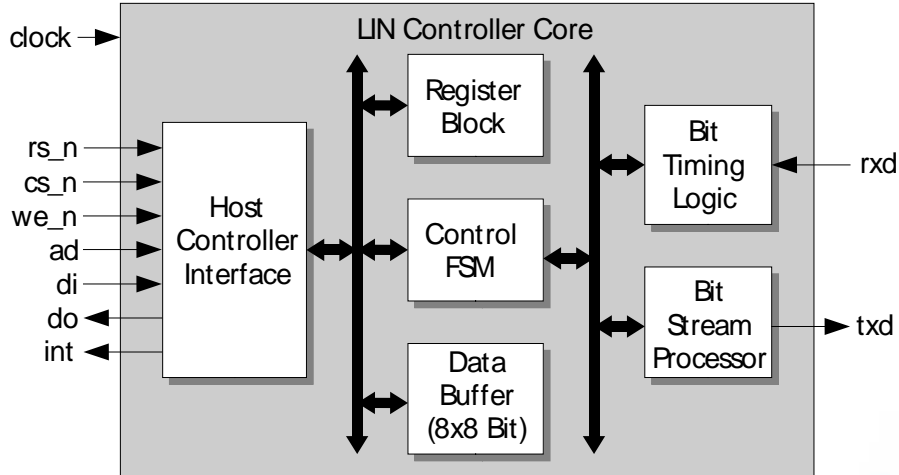
The LIN is a microcode-free design developed for reuse in ASIC and FPGA implementations. The design is strictly synchronous with positive-edge clocking and no internal tri-states.

### Applications

The LIN core can be utilized for a variety of applications including:

- low cost automotive networks
- interfaces for sensors and actuators

### Block Diagram



### Features

- Support of LIN specification 2.1
- Programmable data rate between 1 Kbit/s and 20 Kbit/s
- 8-byte data buffer
- 8-bit host controller interface
- Configurable for support of master or slave functionality
- Slave can be implemented with or without clock synchronization
- Fully synchronous design, available in VHDL or Verilog, completely synthesizable
- The LIN Controller synthesizes to approximate 3000 gates

## Functional Description

The LIN core is partitioned into modules as shown in the block diagram.

### Host Controller Interface

This interface is responsible for handling the communication with the host controller of the system.

### Register Block

The Register Block provides control registers and status registers to control the LIN message transfer. Access to the registers is possible via the host controller interface.

### Data Buffer

The 8-byte Data Buffer stores the data that has to be sent with the current LIN frame or the data that has been received with the last LIN frame. Access to the Data Buffer is possible via the host controller interface.

### Control FSM

The finite control state machine is responsible for the behavior of the core depending on host controller commands and bus activity. It generates and processes the LIN frame fields according to the LIN protocol.

### Bit Stream Processor

This module converts the data stream from parallel to serial (from transmit buffer to bus) and from serial to parallel (from bus to receive buffer).

### Bit Timing Logic

The Bit Timing Logic is responsible for synchronizing the received data stream from the bus with the internal bit time clock.

## Implementation Results

LIN reference designs have been evaluated in a variety of technologies. The following are sample Xilinx results.

Supported Family	Slices <sup>1</sup> master/slave	I/Os <sup>2</sup>	BRAM	Fmax <sup>3</sup> (MHz) master/slave	ISE Version
Spartan-3E 3S1200E-5	325 / 521	28	0	82 / 78	12.2
Spartan-6 6SLX25-3	115 / 209	28	0	139 / 124	12.2
Virtex-4 4VLX15-11	337 / 553	28	0	169 / 141	
Virtex-5 5VLX30-3	120 / 181	28	0	172 / 199	12.2
Virtex-6 6VLX130T-3	89 / 201	28	0	236 / 212	12.2

Notes:

- 1) Optimized for area
- 2) Assuming all core I/Os and clocks are routed off-chip
- 3) Working frequency of LIN controller is 4 MHz
- 4) Slave implemented with clock synchronization

## Core Modifications

The LIN core can be modified to include an acceptance filter. With that, a simple LIN slave that transmits response frames for only one identifier could be realized without host controller.

Please contact CAST, Inc. directly for any required modifications.

## Support

The core as delivered is warranted against defects for ninety days from purchase. Thirty days of phone and email technical support are included, starting with the first interaction. Additional maintenance and support options are available.

## Verification

The core has been verified through extensive simulation and rigorous code coverage measurements

## Deliverables

The core includes everything required for successful implementation:

- VHDL or Verilog RTL source code
- Post-synthesis EDIF (netlist licenses)
- Testbench
- Vectors for testbench
- Simulation and synthesis script
- Expected results for testbench