CAN IP CORE

Features

- Supports CAN 2.0A, and 2.0 B.
- Programmable data rate up to 1 Mbps.
- Technology Independent (ASIC/FPGA).
- Synthesizable Verilog Model.
- Fully synchronous design.
- Parallel processor I/F and optional serial interface.
- Customizable for user requirements.
- IEEE 11898 compliant

Description

CAN2.0B originally developed for the European car industry, is a fast, secure, and cost-effective data bus for multi-master and real-time applications. In addition to automotive applications, it is suitable as a general data bus for industrial control functions.

Example applications of the CAN bus are in the service automation and Textile Machine Industries.

SATYAM created the structured Verilog CAN model for simulation and synthesis for any Target Technology. It can be interfaced via a message filter to various system functions such as Sensor/Activator control, or embedded into a system application interfacing with the microprocessor and various peripheral functions. The core contains the complete data link layer, including the framer, transmit and receive control, error handling, error reporting, and synchronization. Its structured core design and flexible interface enables access to each internal status, error counter, and frame reference.
CAN Controller Signal Description:

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Direction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>reset_n</td>
<td>Input</td>
<td>Active low CAN reset</td>
</tr>
<tr>
<td>phy_clk</td>
<td>Input</td>
<td>CAN clock from Phy (max . 1MHz )</td>
</tr>
<tr>
<td>core_clk</td>
<td>Input</td>
<td>Core clock</td>
</tr>
<tr>
<td>cpu_clk</td>
<td>Input</td>
<td>Clock from cpu</td>
</tr>
<tr>
<td>rd_wr_n</td>
<td>Input</td>
<td>1= cpu read access , when chip_en_n=0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0= cpu write access , when chip_en_n=0</td>
</tr>
<tr>
<td>cpu_addr[7:0]</td>
<td>Input</td>
<td>8 bit address bus from CPU</td>
</tr>
<tr>
<td>cpu_int_ack</td>
<td>input</td>
<td>Interrupt acknowledgement from CPU</td>
</tr>
<tr>
<td>chip_en_n</td>
<td>input</td>
<td>Active low CAN chip enable from CPU</td>
</tr>
<tr>
<td>RxD</td>
<td>Input</td>
<td>Data from CAN Transceiver (Phy)</td>
</tr>
<tr>
<td>TXD</td>
<td>output</td>
<td>Data to CAN Transceiver (Phy)</td>
</tr>
<tr>
<td>cpu_data[7:0]</td>
<td>In/ Out</td>
<td>8 bit data bus between CPU and CAN</td>
</tr>
<tr>
<td>can_int_n</td>
<td>Output</td>
<td>Active low Interrupt signal from CAN</td>
</tr>
<tr>
<td>can_ack</td>
<td>Output</td>
<td>Can acknowledge for CPU access</td>
</tr>
</tbody>
</table>
Sub Level Block Structure of CAN Controller
This section provides a brief overview of the major modules involved in this design.

- Transmitter Block (TXB)
- RAM CTL
- Receiver Block (RXB)
- Processor Interface Block (PIB)
- Tx FIFO
- Rx FIFO

Transmitter Block (TXB)
This block interacts with RAM and Phy, Receiver block. This is responsible for the following functions:

- Construction of frame by adding SOF, RTR bit, reserved bits, CRC, ACK & EOF.
- Initiation of the transmission process after recognizing bus idles.
- Serialization of the frame.
- Insertion of stuff bits (bit stuffing).
- CRC sequence generation.
- Switching to receive mode when the arbitration is lost.
- Error detection (bit error and acknowledgement error).
- Construction and transmission of overload frame.
- Construction and transmission of error frame.

TXB has the following sub blocks:

- Parser (parallel to serial converter)
- Stuffer
- CRCGen
- Error and overload framer.
- Tx Switch.
Receiver Block (RXB)

This block is responsible for the following functions:
- Reception of serial bit stream from the physical layer.
- Bit synchronization.
- De-serialization of the frame structure.
- Deletion of stuff bits (De-stuffing).
- Error detection (CRC, format check, stuff rule check).
- Acceptance filtering.
- Transmission of acknowledgement.
- Recognition of overload condition.

Deliverables

1. Verilog HDL RTL Code (with RS232 Serial Interface)
2. Test Environment in Verilog HDL.
3. Coverage Reports.
4. Logic Synthesis net list.
5. Logic Synthesis Log Files.
7. Application notes.
**Target Device Details**

<table>
<thead>
<tr>
<th>Make</th>
<th>Device</th>
<th>Versatile Count</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actel</td>
<td>M1A3P1000</td>
<td>5601</td>
<td>45.1 MHz</td>
</tr>
</tbody>
</table>

**Contact for requirements:**

SiliconDesign@satyam.com
Satyam Computer Services Ltd
Site: 44, 45, 46-Electronic City
KIADB Industrial Area
Bangalore- 560100
Ph: +91 80 6780 7777