

A Product of the
Cooperative Engineering Program

SAE J1567 AUG87

**Collision Detection
Serial Data
Communications
Multiplex Bus**

SAE Information Report
Issued August 1987

CANCELLED
(Dec 1994)

**S. A. E.
LIBRARY**

No part of this publication may be reproduced in any form,
in an electronic retrieval system or otherwise, without the
prior written permission of the publisher.

Copyright 1987 Society of Automotive Engineers, Inc.

Full document details: SAE J1567_199412



Documents

Document Sharing Hub

COLLISION DETECTION SERIAL
DATA COMMUNICATIONS MULTIPLEX BUS

1. **INTRODUCTION:** The serial communications bus and bus interface special function integrated circuit defined herein was intended to provide a simple, yet reliable, data communications link between members of a distributed processing vehicle multiplex system. The communications protocol chosen minimizes the software support overhead requirement of the modules on the multiplex bus. Appendix A contains a detailed technical summary of the Differential Serial Communications Bus protocol and Interface Integrated Circuit.
2. **PROPOSED PROTOCOL:** The network access method that best meets the requirements is known as CSMA (Carrier Sense Multi Access). A deterministic priority access method of resolving contention, by non-destructive bit-by-bit arbitration, was chosen instead of the nondeterministic random backoff procedure associated with classic collision detection. The message format chosen is shown below and will support a number of higher level protocols. Take note that idle periods are allowed between each byte of data. This permits the use of firmware control and direct connection to the host microcomputer's asynchronous serial I/O port.

(SOM), (id 1), (id 2), (data 1), (data 2) ... (data n), (EOM)

- a - SOM, defines start of message
- b - id 1, 8 bit firmware addressing scheme
- c - id 2, optional identifier
- d - Data, may take any form, that is, data value, CRC, checksum, number of bytes in message, acknowledgement, etc.
- e - EOM, defines end of message

SAE Technical Board Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be reaffirmed, revised, or cancelled. SAE invites your written comments and suggestions.

2. (Continued):

Standard byte synchronization NRZ was chosen as the bit encoding modulation technique. This data byte synchronizing method also provides message integrity by framing error detection. Three bit encoding methods NRZ, PWM, and Bi Phase were originally considered. The choice to use byte sync NRZ encoding as the modulation method was because it is believed to have the potential of providing the best compromise in bandwidth efficiency and data recovery in a noisy vehicle environment while maintaining the required high network data rates without generating excessive EMI.

The first thing considered was the data recovery performance in a noisy vehicle environment. The best performing data decoders should digitally approximate an integration over the bit period. In a noisy environment up to 1/2 of the samples can be corrupted for the NRZ encoding method before data is lost. PWM encoding method suffers a significantly smaller fraction dependent on the duty factor. This reduction in noisy environment performance cannot be compensated by just reducing the data rate.

The second thing considered was the EMI generated by the different data bit encoding method. Some empirical data was collected in a vehicle experiment. The data, although not completely conclusive, indicated that NRZ could operate at a significantly higher bit rate than PWM or Bi Phase encoding methods. It was also felt that the main advantage to using PWM or Bi Phase encoding is that the data bit rate is also encoded in the signal and, therefore, operation can be maintained using a larger clock tolerance. The NRZ requirement for a more precision clock was not a large advantage because all module applications already demanded a crystal or ceramic resonator.

The need for an open network so that minimal effect would result when nodes are added or removed, led to the selection of a broadcast type system. The identification (ID) byte of the message data content is unique and is broadcast across the network. The collision detection hardware within the interface IC determines which transmitting ID's wins without losing bus time when contention occurs. This action satisfies the need for establishing access priority and, therefore, gaining control of the latency for important messages. The message format and the data integrity method is determined by firmware. Maximum flexibility is the result and, thus, the software developer's creativity is not restricted for future applications.

3. FUNCTIONAL DESCRIPTION: Interfacing between a microcomputer and the bus is fairly simple. There are basically three methods; Asynchronous serial, synchronous serial, and parallel to the address and data I/O port. The cost of the parallel interface was avoided by development of a low cost serial interface IC. To make it universally interfaceable to most microcomputers three modes of operation are supported in one device; (1) SCI, (2) SPI, and (3) Buffered SPI.

3. (Continued):

The circuits of the interface IC used when connected to the microcomputer SCI are basic to the operation of all the other modes of serial communications. Therefore, this mode will be explained first. The components of the device (See Fig. 1) include the following:

- a - A contention permitting differential transceiver
- b - A collision detector
- c - An arbitration detector
- d - A digital filter
- e - A bus idle detector
- f - A timing and synchronizing circuit

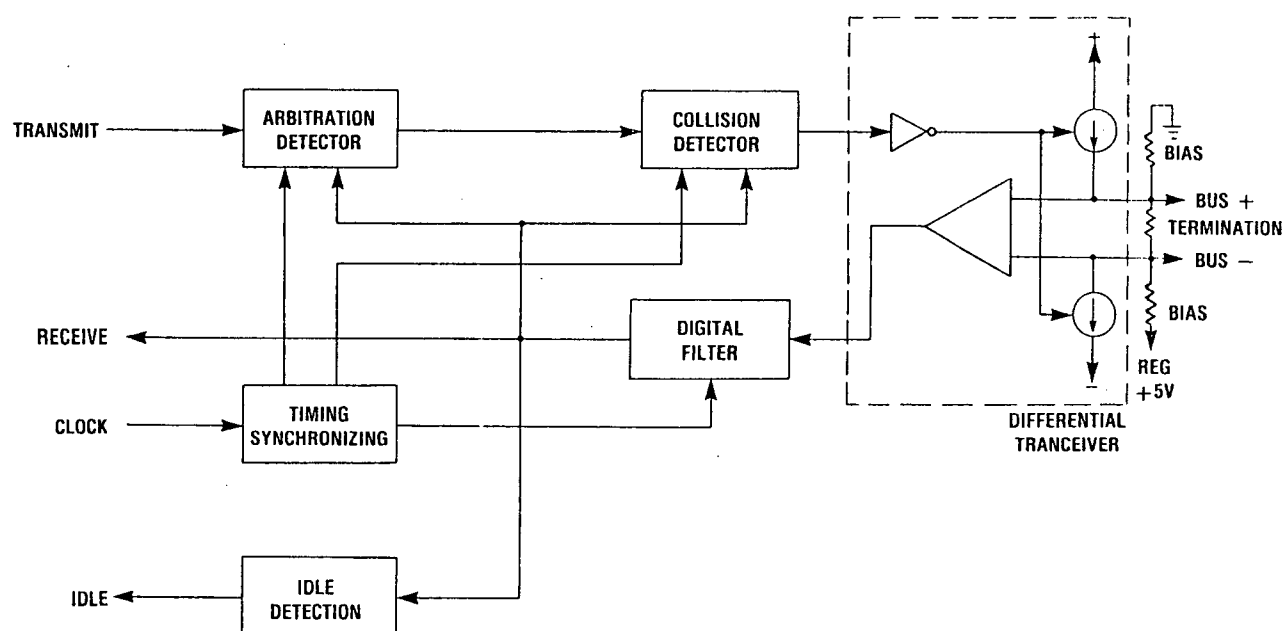


FIG. 1 - SIMPLIFIED BLOCK DIAGRAM OF BUS INTERFACE

- 3.1 Contention Permitting Differential Transceiver: The differential transceiver is a serial interface device which accepts digital signals and translates this information for transmission on a two-wire differential bus. The transmitter section, when transmitting, shall provide matched constant current sources to the bus "+" and bus "-" drivers (Ref. Fig. 1), sourcing and sinking current respectively. When a logic zero is supplied to the "transmit data" input, the differential amplifier shall cause the bus "+" driver to provide source current and the bus "-" driver to provide a matched current sink. A logic one at the "transmit data" input must cause the bus "+" and bus "-" drivers to simultaneously provide a high impedance state.

3.1 (Continued):

The wired "OR" action of the transmitting section allows more than one device to transmit at the same time thus permitting data collisions. The nonsymmetrical action of the bus drivers will allow a transmission of a logic zero, from one device, to overpower the transmission of a logic one from other devices. In this manner two or more devices can simultaneously transmit and contend for the bus, each using a unique message ID byte. The winner is determined by the value (or priority) of the ID byte, without losing bus time. A logic zero bit in one message ID byte has priority over a logic one bit in another message ID byte.

The bus shall depend on external resistor and other components for bias and termination. Clamping diodes may be added to provide a high level of transient protection.

In addition to the transmission of data, the differential transceiver receives data at its bus "+" and bus "-" terminals. The received data is translated back into the standard digital logic levels by a differential amplifier. The microcomputer always receives the actual transmitted data and in this manner can test for loss of arbitration.

- 3.2 Collision Detector: Data collision detection occurs because the transceiver output is reflected back into its input. The data collision detector samples the transmitted signal and the received signal. The timing of this sampling is determined by the timing circuit. When the collision detector determines that a logic one bit is being transmitted, but a logic zero bit is being received, the collision detector blocks the transmitted signal (See waveform shown in Fig. 2). In this way the data collision detector will permit only the interface with the highest priority to continue transmitting. The collision detector action of blocking transmission is also reset by detection of bus idle (10 consecutive idle bits).

3.3 Arbitration Detector:

The arbitration detector works in conjunction with the timing and synchronizing circuit to arbitrate between the start of data to be transmitted with the start of a received message from the bus. The arbitration detector blocks a transmission that could corrupt a message that is already in progress. Also, it allows the device that starts transmission first, after a bus idle, to pass its data through the interface and out onto the bus. In all other nontransmitting devices the arbitration detector blocks transmission of data until the detection of a bus idle condition. When more than one device wants to transmit at about the same time, (greater than 1/4 bit time) the arbitration detector will allow transmission on a first-come, first-serve basis. If data transmission from more than one device on the bus is attempted in near synchronism, (less than or equal 1/4 bit time) the arbitration detector will allow the transmission. However, the data collision detector will permit only the one with the highest priority to continue transmission. The arbitration detector is also reset by the detection of a bus idle.