SAE The Engineering Society For Advancing Mobility Land Sea Air and Space®
INTERNATIONAL
400 Commonwealth Drive, Warrendale, PA 15096-0001

SURFACE VEHICLE RECOMMENDED PRACTICE

SAE J2201

CAN-CELLED JUL1999

Issued 1993-06
Cancelled 1999-07

Superseding J2201 JUN1993

Submitted for recognition as an American National Standard

Universal Interface for OBD II Scan

TABLE OF CONTENTS

QUESTIONS REGARDING THIS DOCUMENT: (724) 772-8512 FAX: (724) 776-0243
TO PLACE A DOCUMENT ORDER: (724) 776-4970 FAX: (724) 776-0790
SAE WEB ADDRESS http://www.sae.org

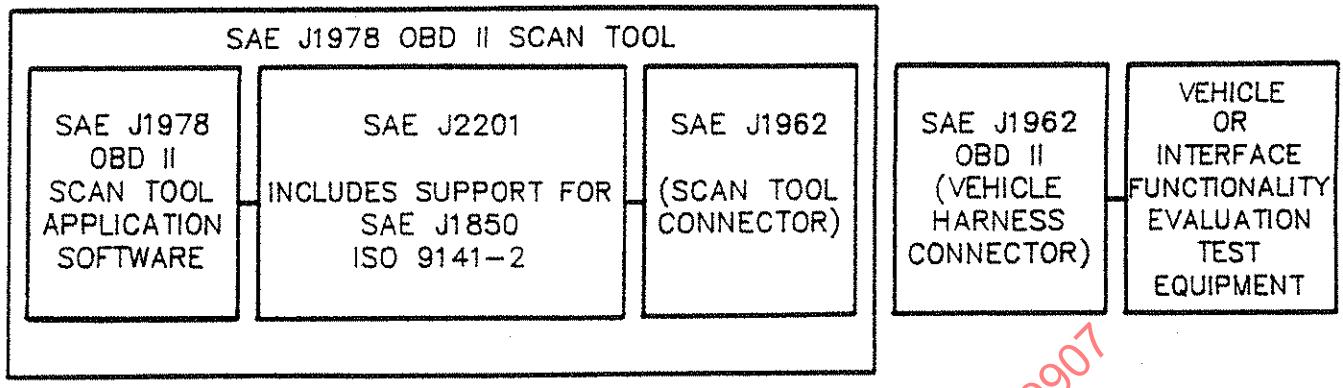
1. ***Scope***—SAE J1978 defines the requirements of the OBD II scan tool. SAE J2201 defines the minimum requirements of the vehicle communications interface for the SAE J1978 OBD II scan tool. This interface connects the SAE J1962 test equipment connector to the hardware/software of the SAE J1978 OBD II scan tool that will use this interface to communicate with vehicles for the purpose of accessing required OBD II functions.

Included in this SAE Recommended Practice are several definitions relating to the interface, and interface functionality evaluation.

Appendix A - Examples include several example interface circuit implementations, which are believed to meet the requirements of this document and of SAE J1978. These examples are NOT requirements of this document. They are provided to assist circuit designers in developing interface circuits.

Appendix B - Supporting Documents includes a list of supporting documents for the examples shown in Appendix A.

1.1 SAE Document Interrelationships—Figure 1 shows the interrelationships between SAE J1978, SAE J1962, and this document.



NOTE—See SAE J1962 Diagnostic Connector for pin assignments.

FIGURE 1—SAE DOCUMENT INTERRELATIONSHIPS

Where any conflict may exist between the requirements contained in SAE J1978 and this document, SAE J1978 is the overriding document.

2. *References*

2.1 Applicable Publications—The terms, definitions, abbreviations, and acronyms contained in SAE J1930 are included by reference.

The following publications form a part of this specification to the extent specified herein. The latest issue of SAE publications shall apply.

2.1.1 SAE PUBLICATIONS—Available from SAE, 400 Commonwealth Drive, Warrendale, PA 15096-0001.

SAE J1850—Class B Data Communication Network Interface
SAE J1930—Electrical/Electronic Terms, Definitions, Abbreviations, and Acronyms
SAE J1962—Diagnostic Connector
SAE J1978—OBD II Scan Tool
SAE J1979—Diagnostic Test Modes
SAE J2205—Expanded Diagnostic Protocol

2.1.2 CALIFORNIA AIR RESOURCE BOARD PUBLICATIONS—Available from California Air Resource Board Documents

California Code of Regulation, Title 13, 1968.1—Malfunction and Diagnostic System Requirements - 1994 and Subsequent Model-Year Passenger Cars, Light-Duty Trucks, and Medium-Duty Vehicles With Feedback Fuel Control Systems

2.1.3 EPA REGULATIONS—Available from EPA

Federal Register Tuesday September 24, 1991, Part II Environmental Protection Agency 40 CFR Part 86—Air Pollution Control; New Motor Vehicles and Engines: On-Board Diagnostic Systems on 1994 and Later Model Year Light-Duty Vehicles and Light-Duty Trucks; Proposed Rule

2.1.4 ISO PUBLICATIONS—Available from ANSI, 11 West 42nd Street, New York, NY 10036-8002.

ISO 9141-2—Road vehicles—Diagnostic systems—CARB requirements for interchange of digital information ISO/TC 22/SC 3/WG 1 - N 425 E/REV April 1991

3. Definitions

3.1 Application Software—As used in this document, this term refers to the microprocessor programming that controls the external equipment/scan tool hardware so as to perform required SAE J1978 OBD II Scan Tool functions.

3.2 DLCS—(Data Link Controller Serial) refers to the integrated circuit family developed by General Motors that supports SAE J1850 10.4 Kbps VPW CRC communication.

3.3 HBCC—(Hosted Bus Controller Circuit) refers to the integrated circuit developed by Ford Motor Co. that supports SAE J1850 41.6 Kbps PWM CRC communication.

3.4 OBD II—(On Board Diagnostics II) common term that refers to the requirements of the California legislation-California Code of Regulation, Title 13, 1968.1.

3.5 PCI—(Programmable Communications Interface) refers to a set of integrated circuits which includes a control microcomputer, the Symbol Encoder Decoder (SED) and the Integrated Driver Receiver (IDR). The PCI was developed by Chrysler Corporation and supports SAE J1850 10.4 VPW communication with both CRC and Checksum error techniques.

3.6 JCI—(SAE J1850 Communications Interface) refers to an integrated circuit manufactured by Motorola which supports both SAE J1850 10.4 VPW with CRC and SAE J1850 41.6 PWM with CRC communication.

4. Acronyms and Abbreviations

CRC - Cyclic Redundancy Check
CS - Checksum
EDP - Enhanced Diagnostic Protocol (SAE J2205)
IBS - Inter-Byte Separation
IFR - In-Frame Response
NRZ - Non Return to Zero
PWM - Pulse Width Modulation
SCI - Serial Communications Interface
SPI - Serial Peripheral Interface
UART - Universal Asynchronous Receiver/Transmitter
VDD - 5.0 VDC from Regulated Power Supply
VPW - Variable Pulse Width Modulation
"~" - Indicates Active Low Signals

*5. **Requirements***—This section defines the required message structure support, signal ground, chassis ground, minimum scan tool connector cable length, and other requirements for the interface to be used by an SAE J1978 OBD II scan tool.

5.1 Interface and Message Protocol Support—The interface defined in this document must support the interface requirements and message protocol requirements of SAE J1978.

5.2 In-Frame Response—When a single byte in-frame response (IFR) is required during the reception of a 41.6 Kbps PWM SAE J1979 message, the interface will support the transmission of the node address as a single byte IFR.

5.3 **Signal Ground**—The Signal Ground pin of the scan tool side of the SAE J1962 diagnostic connector must be used as the signal ground reference for all interface transceivers required by this document.

5.4 **Maximum Voltage Differentials**—The interface described in this document includes any required interface transceivers and the cabling connecting the interface transceivers to the scan tool side of the SAE J1962 connector. Any interface connected to a vehicle through the SAE J1962 diagnostic connector is considered a part of the vehicle network and must operate within the limits of that network, which are described in SAE J1850 and ISO 9141-2.

The maximum voltage differential, e.g., due to load current, noise, etc., between any two nodes of a vehicle data communication network, where the interface described in this document is considered as a node on the vehicle network when connected to the vehicle, must be less than the limits specified in SAE J1850 and ISO 9141-2.

The following are maximum voltage differential values for the interface described in this document (as measured between the signal ground connection of all interface transceivers and the Signal Ground pin on the vehicle side of the SAE J1962 diagnostic connector) are:

0.25 V peak noise
0.1 VDC offset

5.5 **Chassis Ground**—The Chassis Ground pin of the scan tool side of the SAE J1962 diagnostic connector is available for any use, with the exception that the minimum DC impedance between the Chassis Ground and the Signal Ground is 1 MΩ.

5.6 **Minimum Connector Cable Length**—The minimum length between the ground reference connection of each interface transceiver and the Signal Ground connection of the vehicle side of the SAE J1962 diagnostic connector is 2 m.

5.7 **Other Requirements**—The interface must support the requirements of SAE J1850, SAE J1962, SAE J1978, SAE J1979, SAE J2205, and ISO 9141-2.

6. ***Interface Functionality Evaluations***—The functionality of any proposed interface implementations must be evaluated by either of the following:

a. The use of multiplex bus interface equipment from I + ME, or equivalent equipment, that with appropriate software simulates vehicles that use the required implementations of SAE J1850 and ISO 9141-2, or
b. The use of representative vehicles of motor vehicle manufacturers that use the required implementations of SAE J1850 and ISO 9141-2.

PREPARED BY THE SAE VEHICLE E/E SYSTEM DIAGNOSTICS STANDARDS COMMITTEE

APPENDIX A

EXAMPLES

A.1 General Example Information—This section shows example interface implementations that are believed to meet the requirements of this document.

These examples are NOT requirements of this document. These examples are intended as an assist to interface circuit designers.

Table A1 illustrates a summary of some of the capabilities available with these examples that are over and above the requirements of this document and some that are required.

TABLE A1—SUMMARY OF CAPABILITIES FOR THE FOLLOWING EXAMPLES

	DLCS & HBCC Section A.2	PCI & HBCC Section A.3	ST9 Section A.4	F690 & F691 Section A.5	JCI Section A.6
Error Checking	CRC	CS & CRC	CS & CRC	CS & CRC	CRC
Comprehends IBS(1)	No	Yes	Yes	Yes	No
Header Bytes	3	1 & 3	1 & 3	1 & 3	3
Generate Break	Yes	Yes	Yes	Yes	Yes
Control Function(2)	No	Yes	Yes	No	No
Host Required	Yes	Optional(3)	Optional(3)	Yes	Yes

1. IBS is part of only 10.4 Kbps VPW networks. It is not a part of 41.6 Kbps PWM networks, nor a part of ISO 9141–2 networks.
2. Control Function refers to parts of an interface that are implemented in software and executed by some form of microcomputer.
3. Host Required refers to the need for a host microcomputer to perform the functionality of SAE J1978. Some interface implementations include a microcomputer as a part of the interface. In some of these implementations, this microcomputer may have enough excess capability so as to also be able to perform "host microcomputer" functions.

Error checking (CS or CRC), IBS, header bytes, and the Break signal are defined in SAE J1850.

A.1.1 Transient Protection—While some of the examples may show transient protection, this document does not specify or require transient protection.

A.1.2 Host Support Not Included—The examples shown here support a nominal type of host interface and are not meant to be directly applicable to any particular host microprocessor, host interface, and/or host software. The host application software (e.g., interface drivers, message building routines, message processing routines, etc.) and/or additional hardware (e.g., signal buffers, signal inverters, clock synchronization circuits, etc.) required to complete the interface to the example circuits shown in this document is the responsibility of the designer/implementer and is not shown here.

A.1.3 Supporting Documents—Appendix B - Supporting Documents identifies documentation that may be used to aid in the understanding of the examples shown in this document.

A.1.4 Common ISO 9141-2 Support—Many of the examples include a common implementation for ISO 9141-2. This common implementation is described here (see Figure 2-D ISO 9141-2).

The ISO 9141-2 document describes the requirements of the ISO 9141-2 interface. This interface is an asynchronous serial communication link using NRZ bit encoding. A 5 bps bit rate is used during module communication initialization and a 10.4 Kbps bit rate is used for all further communications.

This interface includes a K line which is bidirectional and used for all communication phases, and a L line which is output only from the scan tool and is only used during the 5 bps module communication initialization phase. When used to transmit data to a vehicle, both the L and K lines are driven to either ground or battery level, through a 510 Ω resistor, depending on the data value. When used to receive data from a vehicle, the K line is referenced to 1/2 the battery voltage, with a small amount of hysteresis for noise immunity, to determine the logic value of the data being transmitted by a vehicle module.

In all of the examples, the L and K lines are connected to the SCI port of either the host or a control microprocessor. A select signal from the host/control microprocessor is used to enable the L line when necessary.

A.1.5 Electromagnetic Compatibility—The Electromagnetic Compatibility (EMC) characteristics of these examples have not been investigated nor estimated. The ability of the examples to meet EMC or noise radiation requirements is unknown.

A.1.6 No Responsibility Assumed by Contributors—No responsibility is assumed by Chrysler Corporation, Ford Motor Co., General Motors Corporation, Delco Electronics Corporation, SAE, SGS-Thomson, Silicon Systems, Motorola, or any other company for use of any of the examples or products shown in this document nor for any infringements of patents and trademarks or other rights of other parties resulting from the use of these examples or products. No license is granted under any patents, patent rights, or trademarks of the previously named parties.

A.1.7 Additional Capabilities of Examples—Many of the examples support capabilities beyond the requirements of this document. Such capabilities are discussed in a separate section of each example.

A.2 DLCS, HBCC, and ISO 9141-2 Interface Example—This example (see Figure A1a) combines the use of the Data Link Controller Serial (DLCS) integrated circuit developed by Delco Electronics, the Hosted Bus Controller Circuit (HBCC) integrated circuit developed by Ford, a discrete 41.6 Kbps transmit driver, and the common ISO 9141-2 interface to provide the required interface support. (Figure A1b shows the detail of the DLCS, Figure A1c shows the detail of the HBCC with its Transmit Driver, and Figure A1d shows the detail for the common ISO 9141-2.)

The DLCS and HBCC communicate with the host through the host's SPI port, while the ISO 9141-2 interface communicates with the host through the host's SCI port. Separate logic interface lines are used by the host to individually select communication between the host and either the DLCS and HBCC, and to enable the L line.

A.2.1 General Overview—The DLCS and HBCC perform as communication peripheral integrated circuits. Following initialization, they isolate the host from bus communications tasks and are only serviced by the host when the host is loading a message for transmission or unloading a received message. The host receives an interrupt when a received message is available and when message transmission is completed. The host only handles communications as full messages. Several control and status registers are available in both the DLCS and HBCC to control their operation and indicate the status of their communications tasks.

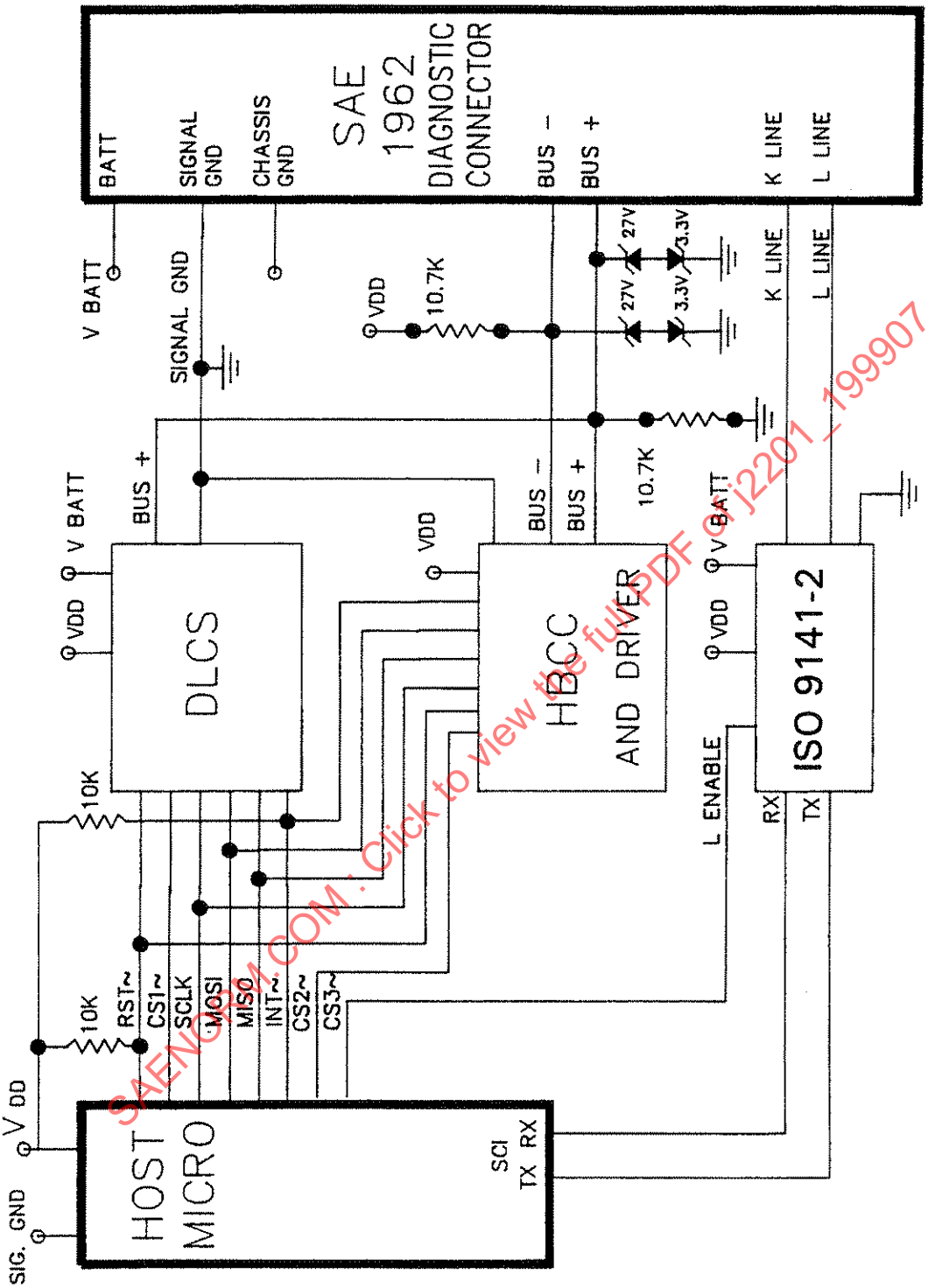


FIGURE A1A—DLCS, HBCC, AND ISO 9141-2 EXAMPLE

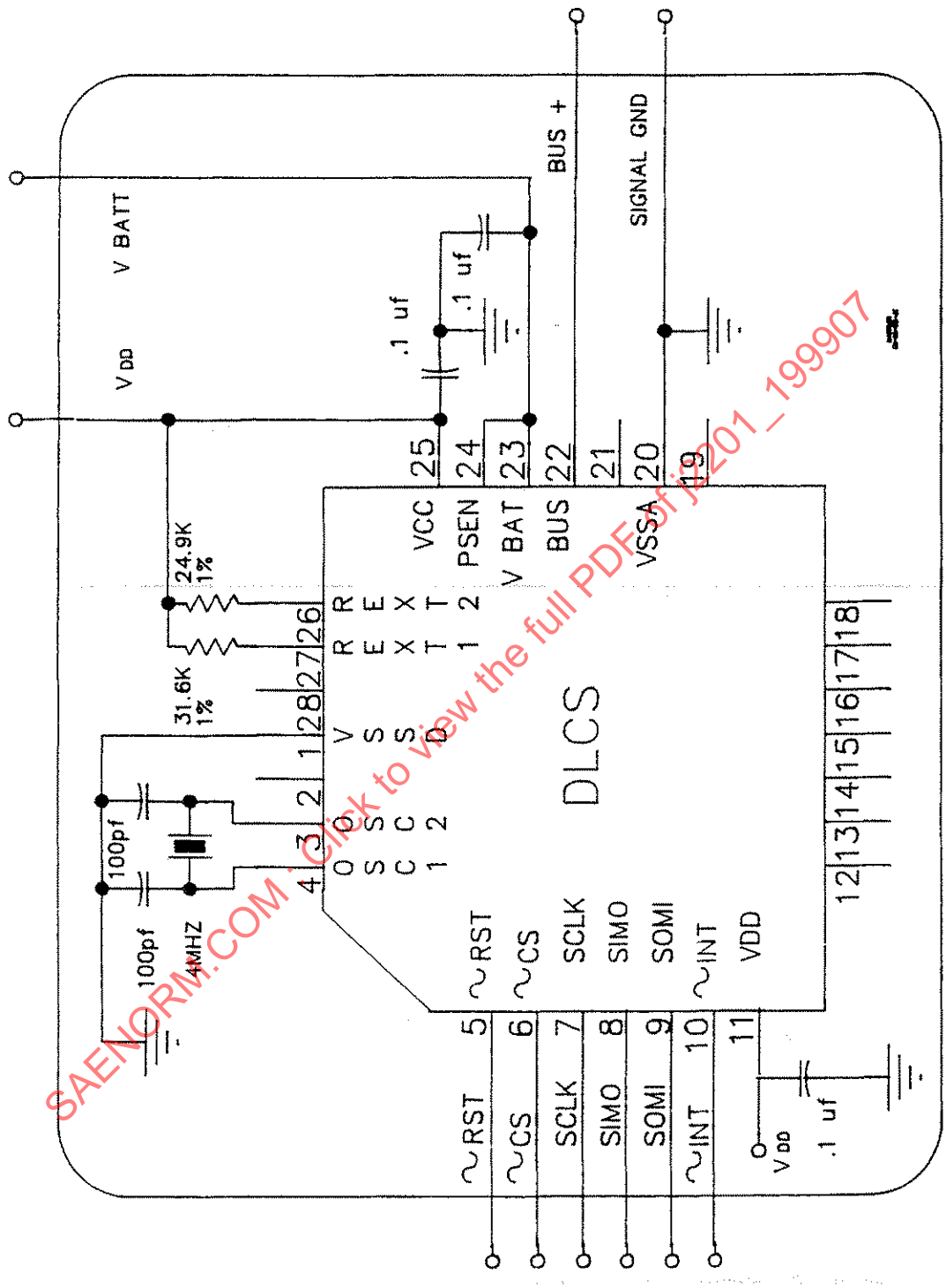


FIGURE A1B—DLCS DETAIL

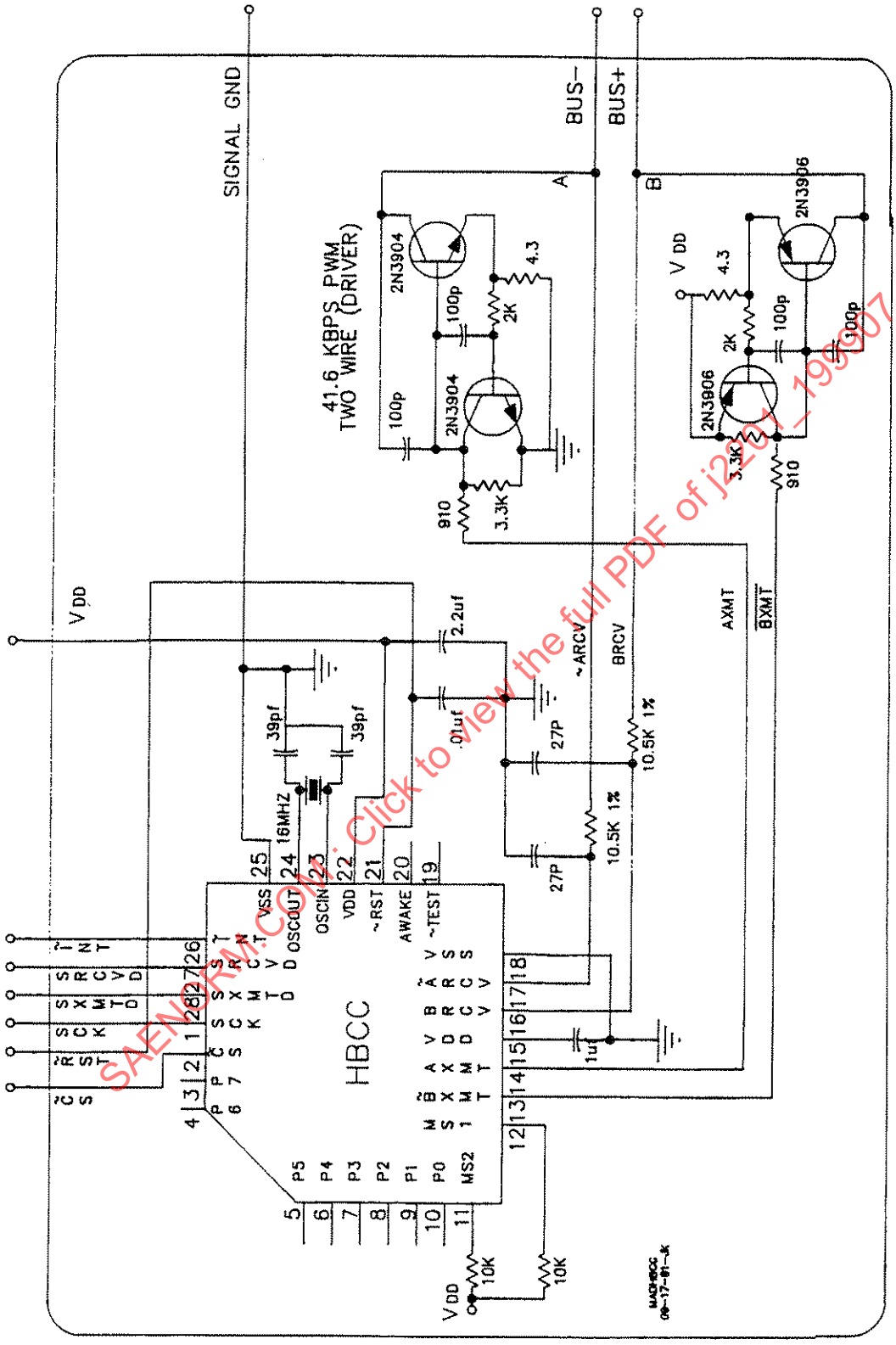


FIGURE A1C—HBCC AND TRANSMIT DRIVER DETAIL

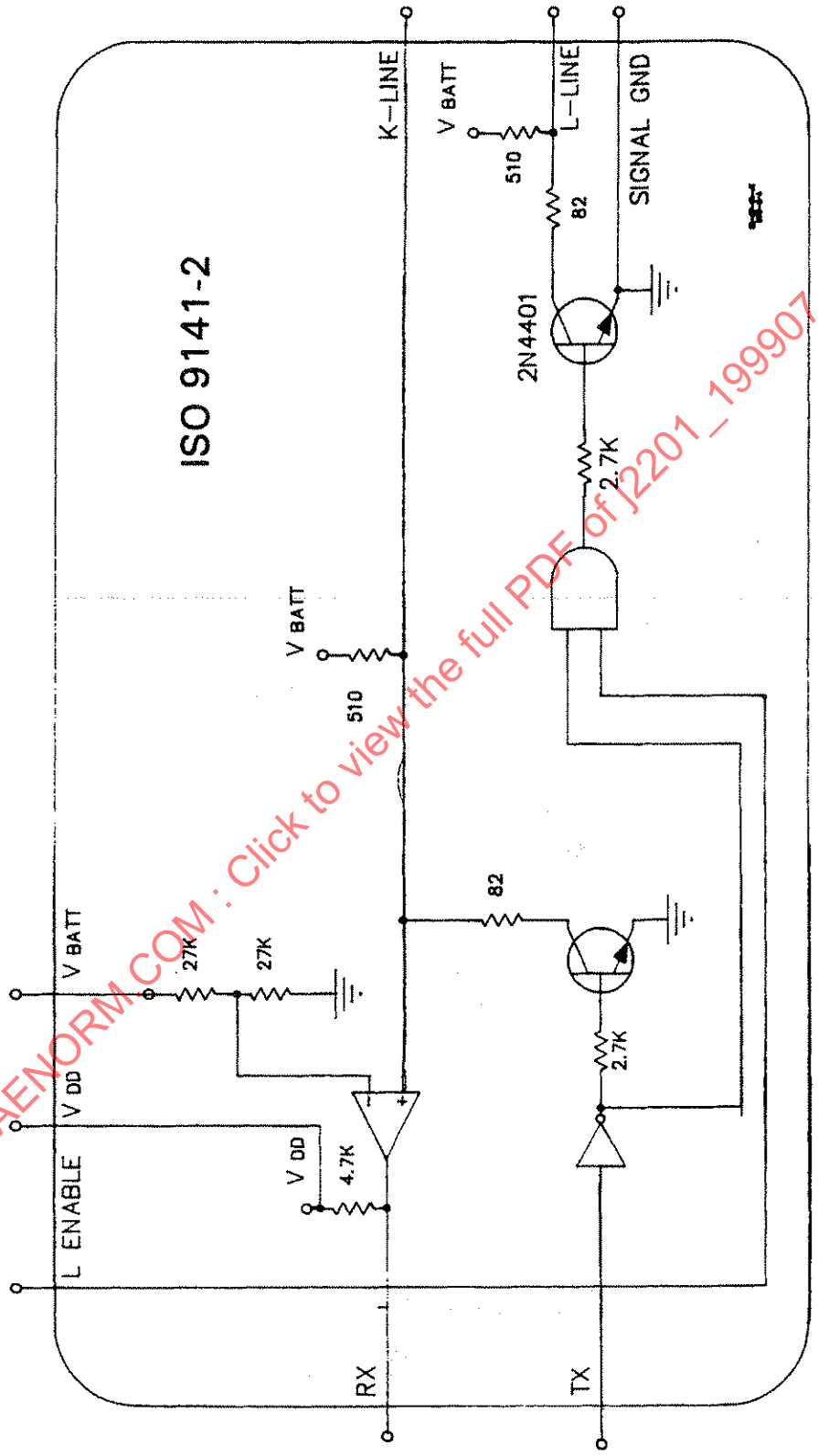


FIGURE A1D—ISO 9141-2 DETAILS

This example uses chip select lines from the host to select which peripheral the host wishes to communicate with. Control data from the host enables and disables the DLCS and HBCC interfaces.

Because these devices are available from semiconductor suppliers who are also supplying them to major automotive manufacturers for use in production vehicles, compatibility with those vehicles should be readily verifiable.

The ISO 9141-2 interface is as described in A.1.4. The host uses its SCI interface to communicate with a vehicle through the ISO 9141-2 interface. A logic line is used by the host to enable the L line when necessary.

A.2.2 DLCS Operation—The DLCS contains a logic section and an analog section in a single integrated circuit. The logic section includes the SPI interface to the host, a transmit buffer, a receive buffer, status and control registers, bit timing, and symbol encoding and decoding logic. The analog section includes the transceiver circuitry. Data transfer between the host and the DLCS is done with two byte SPI transfers. The host sends to the DLCS a command byte and a data byte, while the DLCS sends to the host a status byte and a data byte. The content of the data bytes is dependent on the command.

a. Commands sent to the DLCS include:

1. General commands (e.g., send break, go to sleep, terminate transmit retry, load configuration data)
2. Transmit commands (e.g., load transmit data, transmit message)
3. Receive commands (e.g., unload receive data, flush byte, flush message)

b. Status data send from the DLCS indicate the following:

1. Receive status (e.g., buffer contains bytes or messages)
2. Transmit status (e.g., buffer contains bytes or messages)
3. Data link status (e.g., data link shorted)

Operation of the DLCS consists primarily of three functions: initialization, transmission of a message, and reception of a message. Each of these will be discussed in more detail as follows:

A.2.2.1 INITIALIZATION—Initialization is accomplished by sending the DLCS a command byte to load the accompanying data from the host as configuration data. The DLCS can be configured to enable or disable it to interrupt the host, thereby allowing either polling or interrupt based signalling schemes to be used, and to configure the oscillator divisor. Once the DLCS is initialized, normal transmission and reception can begin.

A.2.2.2 MESSAGE TRANSMISSION—Loading a message from the host to the DLCS for transmission on the bus is done by several 2 byte SPI transfers. Each transfer contains a command byte to load the accompanying data as transmit data. Loading the first and last byte of the message must be done with specific command bytes. Once the command to load the accompanying data as the last transmit data byte is received by the DLCS, the DLCS will begin an attempt to transmit the message on the bus. With each 2 byte SPI transfer with the DLCS, a status byte and data byte will be transferred to the host. The status byte is used to determine the status of the transmit buffer, i.e., empty, contains some bytes, full, etc. The status byte also contains receive status. When loading a transmit message into the DLCS, the data bytes returned by the DLCS to the host are ignored.

The DLCS handles all data link access responsibilities, i.e., idle bus detection, arbitration, automatic retry (if enabled), etc.

A.2.2.3 MESSAGE RECEPTION—The DLCS on its own receives and buffers a complete message from the bus, checks the received CRC, sets a status flag indicating a message has been received and if any errors occurred and, if enabled, interrupts the host. Retrieving a message received by the DLCS is done first by either polling the DLCS to determine if a message has been received, or by servicing an enabled interrupt that occurs upon reception of a message. (The remainder of this discussion assumes the interrupt method is used.)

Upon receiving an interrupt, performing a 2 byte SPI transfer with the DLCS will result in the transfer of status and data to the host. The status byte is used to determine if the receive buffer contains any bytes of a message or a complete message(s), and the status of the transmit buffer. Once all the bytes of a message have been received by the DLCS an indicator will be set in the status byte and, if enabled, an interrupt will be generated for the host. The status byte will also indicate whether the message was received correctly or that an error occurred, i.e., CRC error, incomplete byte, and bit timing error. If the message unloaded from the DLCS was received while the DLCS was attempting to transmit a message, the status byte will also indicate transmission status values such as overrun or underrun and whether bus arbitration was won or lost.

A.2.3 DLCS Pin Names and Descriptions—The names and descriptions of the pins of the DLCS are shown in Table A2.

TABLE A2—DLCS PIN NAMES AND DESCRIPTIONS

Pin Number	Pin Name	Description
1	Vssd	Digital Ground
2	LOTI	Logic Out Transmitter In (Test)
3	OSC2	Oscillator 2
4	OSC1	Oscillator 1
5	RST~	Reset~ (Active Low)
6	CS~	Chip Select ~ (Active Low)
7	SCLK	SPI Serial Clock
8	SIMO	SPI Slave In Master Out
9	SOMI	SPI Slave Out Master In
10	INT~	Interrupt~ (Active Low)
11	Vdd	Digital Voltage Supply (+5 V)
12	N/A	Not Used
13	N/A	Not Used
14	N/A	Not Used
15	N/A	Not Used
16	N/A	Not Used
17	N/A	Not Used
18	N/A	Not Used
19	N/A	Not Used
20	Vssa	Analog Ground
21	LOAD	Bus Load
22	BUS	Bus Output
23	Vbatt	Battery Voltage
24	PSEN	Power Supply Enable
25	Vcc	Analog Voltage Supply (+5 V)
26	REXT2	External Resistor 2
27	REXT1	External Resistor 1
28	LITO	Login In Transmitter Out (Test)

A.2.4 HBCC Operation—The HBCC contains a logic section and an analog section in a single integrated circuit. The logic section of the HBCC supports two types of serial and two parallel host interfaces. This allows for the HBCC to interface with standard Intel or Motorola microprocessors or others which have compatible ports. For this example, the Motorola SPI interface has been used. The logic section also includes a transmit buffer, a receive buffer, status and control registers, bit timing, symbol encoding, and decoding logic. The analog section is limited to the receiver circuitry. The transmitter driver circuit and receiver input noise filter are implemented by external discrete devices.

Data transfer between a host and the HBCC is done with 2 byte SPI transfers. The host transfers an indirect address control byte (ACB) and a data byte to the HBCC. Generally the indirect address points to one of the internal registers of the HBCC and the data sent from the host is stored in the addressed register. During a 2 byte SPI transfer, the host receives a status byte and a data byte from the HBCC. Special address conventions also allow for block type data transfers between the host and the HBCC message buffers.

a. Commands sent to the HBCC include:

1. General commands (e.g., initialize node address)
2. Transmit commands (e.g., load transmit buffer)
3. Receive commands (e.g., read receive buffer)

b. Status data received from the HBCC include the following:

1. Receive status (e.g., received byte count)
2. Transmit status (e.g., transmission completed OK)
3. Network wire status (e.g., network wire shorted)

The operation of the HBCC consists primarily of three functions: initialization, transmission of a message, and reception of a message. Each of these will be discussed in more detail as follows:

A.2.4.1 INITIALIZATION—Initialization is accomplished by sending the HBCC commands to load the accompanying data as configuration information. The HBCC can be configured to enable or disable interrupts to the host, thereby allowing a polling scheme to be used, and to select the bus bit rate. Receive message filter look up tables can also be loaded to screen incoming messages, thus reducing host burden. Once the HBCC is initialized, normal transmission and reception can begin.

A.2.4.2 SENDING A MESSAGE—Loading a message into the HBCC for transmission on the bus is done by several SPI transfers. Each transfer contains an indirect address byte which defines where to store the accompanying data byte into the HBCC transmit buffer. Once the entire message has been loaded into the transmit buffer, the command to transmit the message is sent to the HBCC. The HBCC will then on its own begin to synchronize with the activity on the bus and transmit the message on the bus. The HBCC handles all data link access responsibilities, i.e., idle bus detection, arbitration, and automatic retry. When a message transmit attempt is completed, an interrupt is generated and the host can check the resultant status, including the status of an inframe response.

A.2.4.3 RECEIVING A MESSAGE—Receiving bus messages using an HBCC is generally based on servicing the interrupt that occurs, if enabled, at the completion of message reception. The HBCC can be configured with interrupts either enabled or disabled, but for this example, interrupts are assumed to be enabled. Upon receiving an interrupt, the host will perform a two byte SPI transfer and receive from the HBCC a status byte and a data byte. The status byte can be used to determine how full the receive buffer is, and whether certain receive errors have occurred. A received message can subsequently be transferred from the HBCC to the host.

A.2.5 HBCC Pin Descriptions—The names and descriptions of the pins of the HBCC are shown in Table A3.

TABLE A3—HBCC PIN NAMES AND DESCRIPTIONS

Pin Number	Pin Name	Description
1	SCLK	Serial Clock
2	CS ~	Chip Select ~ (Active Low)
3	USER7	User Input 7
4	USER6	User Input 6
5	USER5	User Input 5
6	USER4	User Input 4
7	USER3	User Input 3
8	USER2	User Input 2
9	USER1	User Input 1
10	USER0	User Input 0
11	MS2	Mode Select 2
12	MS1	Mode Select 1
13	BXMT ~	Network B Drive~ (Active Low)
14	AXMT	Network A Drive
15	Vdd	Analog Voltage Supply (+5 V)
16	BRCV	Network B Receive
17	ARCV~	Network A Receive (Active Low)
18	GND	Analog Ground
19	TESTENA ~	Test Enable A (Active Low)
20	AWAKE	HBCC Awake
21	RST ~	Reset ~ (Active Low)
22	Vdd	Digital Voltage Supply (+ 5 V)
23	OSCIN	Oscillator 1
24	OSCOUT	Oscillator 2
25	GND	Digital Ground
26	INT ~	Interrupt~ (Active Low)
27	SRCVD/SDAT	Serial Data Receive/Serial Data I/O
28	SXMTD/DIR	Serial Data Transmit/Data Direction

A.2.6 Additional Capabilities—Both the DLCS and HBCC include support for both transmit and receive message buffering and automatic message retransmission. The HBCC also includes support for receive message filtering, wake up, and 41.6 Kbps PWM bus physical layer fault tolerance. The DLCS also includes support to send and receive Break and for wake up.

A.3 PCI, HBCC, and ISO 9141-2 Interface Example—This example, (see Figure A2a) combines the use of the Programmable Communication Interface (PCI) developed by Chrysler, the Hosted Bus Communication Controller (HBCC) developed by Ford, a discrete 41.6 Kbps transmit driver, and the common ISO 9141-2 interface to provide the required interface support. (Figure A2B shows the detail of the PCI.)

A description of the HBCC interface is found in A.2.4.

A description of the ISO 9141-2 interface is found in A.1.4.

A description of the PCI is found in this section.

The host communicates with the PCI and the HBCC through the host's SPI interface. The host operates the SPI interface as the master and uses logic control lines to select either the PCI or the HBCC to communicate with. Control information sent from the host to the PCI and the HBCC cause the PCI and HBCC to be enabled or not. The ISO 9141-2 interface uses the SCI interface of the host and a logic line to enable the L line.

A.3.1 General Overview—The PCI operates at a 10.4 Kbps data rate using the symbols specified in SAE J1850.

The PCI is controlled by the coordinated interdependence of both hardware and software. The system consists of the interconnection of three devices:

a. A Control Microcomputer with custom software
b. A Symbol Encoder/Decoder (SED)
c. An Integrated Driver Receiver (IDR)

The control microprocessor controls the operation of the SED and interfaces with the host.

The SED is a digital gate array device which performs symbol encoding and decoding, timing and bus synchronization.

The IDR is an analog ASIC which performs as a 10.4 Kbps VPW bus transceiver for the PCI.

A.3.2 Control Microcomputer—The control microcomputer has overall control of the PCI's operation and interfaces with the host and the SED. The control microcomputer, under software control, interprets host commands and bus symbols.

A.3.2.1 CONTROL MICROCOMPUTER SOFTWARE STRUCTURE—The control microcomputer's most critical task is to service the IRQ interrupts generated by the SED. Less critical tasks are handled on an as-needed basis by the interruptable main program.

The IRQ interrupt processing task services the following:

a. Transmission of Frame Symbols
b. Reception of Frame Symbols
c. Bit-By-Bit Arbitration
d. Error Detection
e. Symbol encoding and decoding
f. In-frame response

The main program services the following:

a. Interface to the host

1. SPI (or Parallel) interface service
2. Chip selection of control microcomputer by the host

b. Bus calculations

1. Checksum or CRC
2. Transmit and receive message buffering
3. Error interpretation

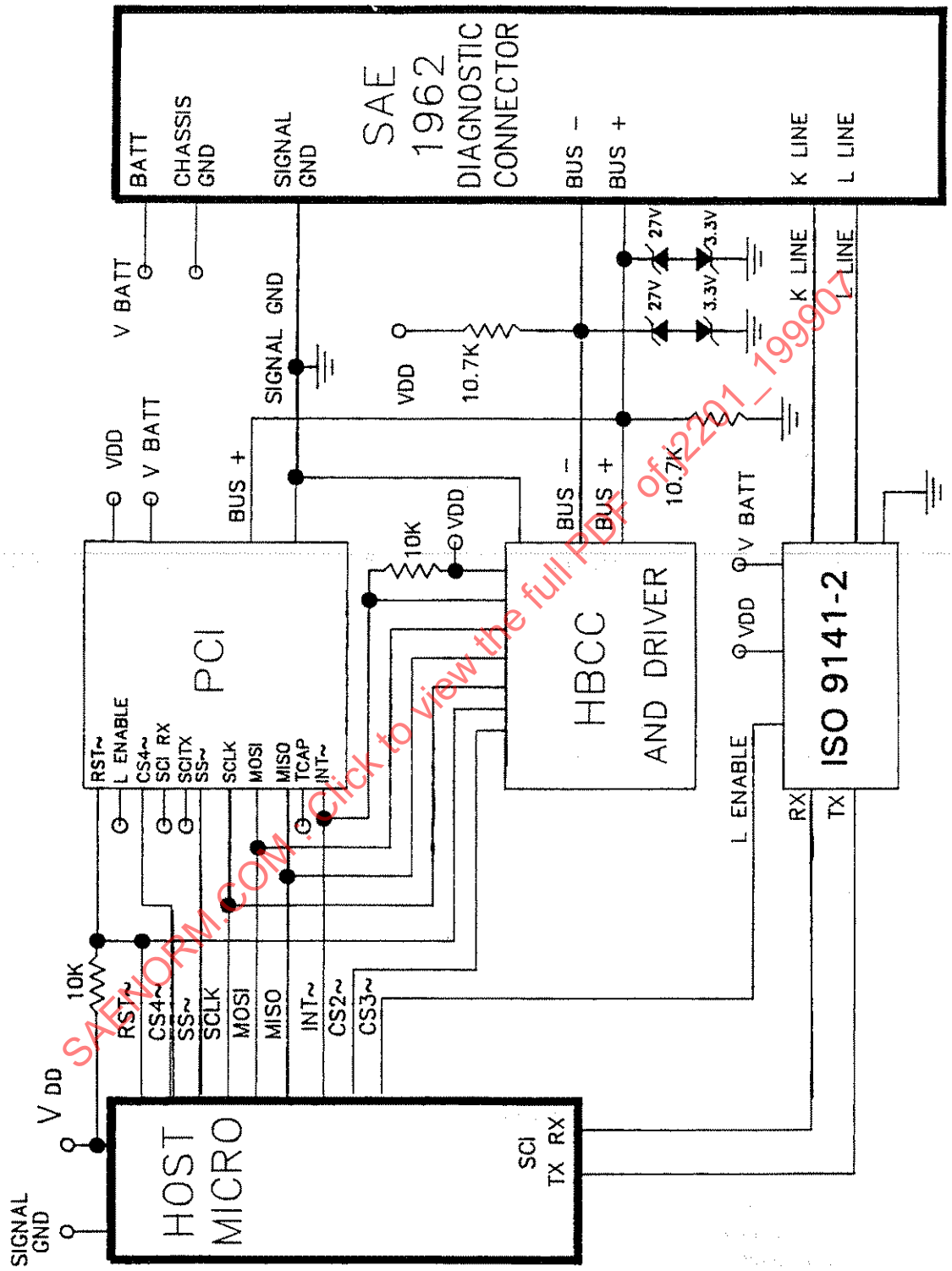


FIGURE A2A—PCI, HBCC, AND ISO 9141-2

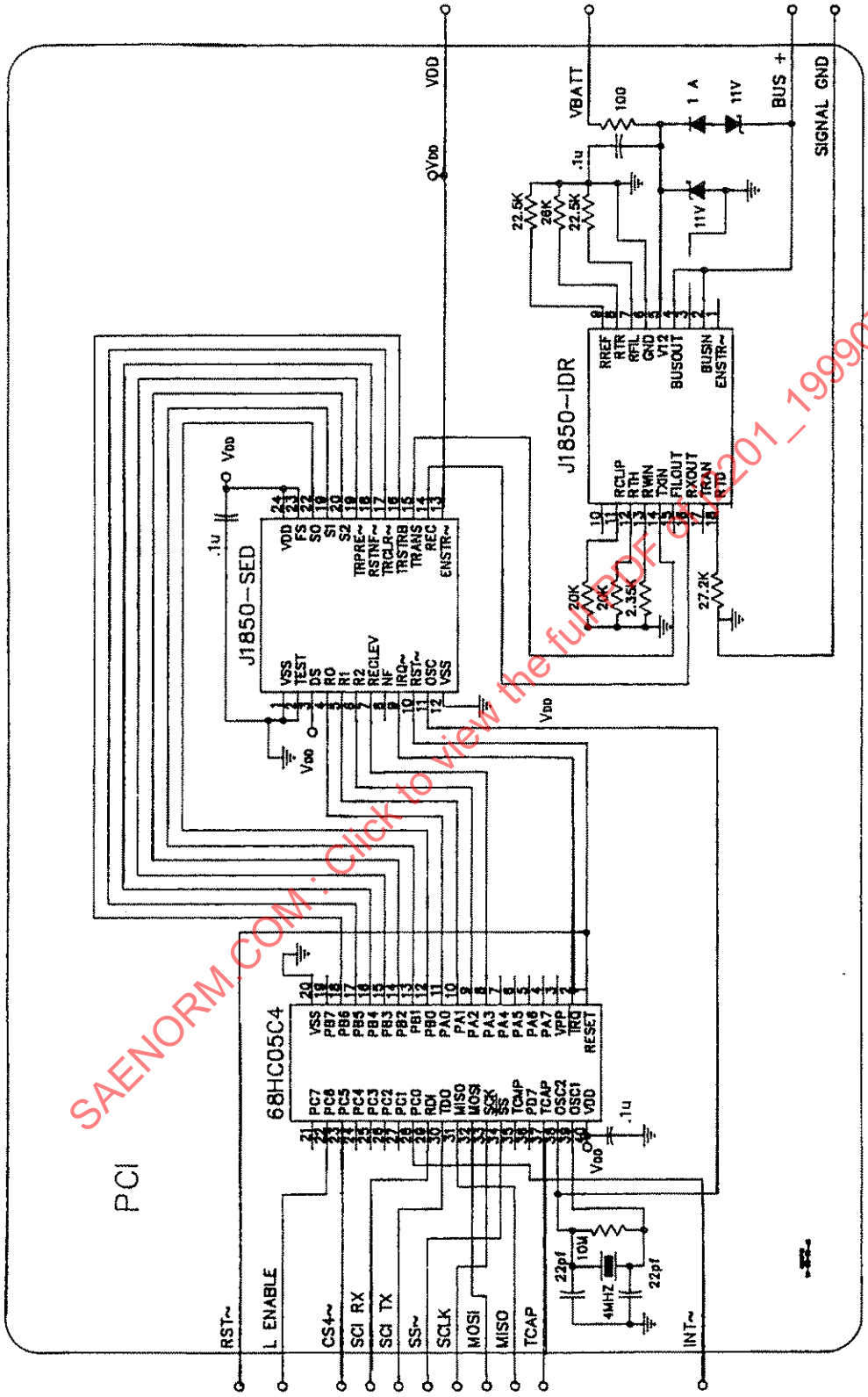


FIGURE A2B—PCI

a. Optional tasks when operating as host also

1. Keyboard inputs
2. Display drivers

A.3.2.2 CONTROL MICROCOMPUTER INTERFACE TO THE HOST MICROCOMPUTER—The PCI supports both a SPI or a parallel interface with the host. The host functions as interface master for both types of interfaces.

A.3.2.3 CONTROL MICROCOMPUTER INTERFACE TO THE SED—The interface between the control microcomputer and the SED shall be as defined by A.3.3 and Table A4.

TABLE A4—CONTROL MICROCOMPUTER AND SED INTERCONNECTION

Input Port	SED Output	Output Port	SED Input
PA0	R0	PB0	S0
PA1	R1	PB1	S1
PA2	R2	PB2	S2
PA3	RECLEV	PB3	TRPRE~
PA4	NF	PB4	RSTNF
IRQ~	IRQ~	PB5	TRCLR~
		PB6	TRSTRB

A.3.3 Symbol Encoder Decoder (SED)—The SED receives commands from the control microcomputer and converts these commands to SAE J1850 VPW symbols on a symbol-by-symbol bit-by-bit basis. Reception of messages is similarly accomplished, with the SED converting each received symbol data that is fed to the control microcomputer for deciphering.

A.3.3.1 SYMBOL ENCODER—The control microcomputer initiates the transmission of a frame by first strobing Transmit Preset TRPRE ~ to a logic "0" and strobing Transmit Strobe TRSTRB to a logic "1". The S0, S1, and S2 Inputs are then set to the logic levels defined in Table A5 in order to activate the transmit symbol generator circuit. During each symbol, the Transmit Strobe TRSTRB is strobed. The transmit symbol genenerator produces the required symbol on the Transmit TRANS Output of the SED. When the symbol has completed transmission, the SED Receive Circuit generates an IRQ ~ interrupt to the control microcomputer to obtain the next symbol. This cycle is repeated until the entire frame is transmitted.

TABLE A5—TRANSIT SYMBOL DEFINITION

Inputs S2	Inputs S1	Inputs S0	Output Description J1850 Xmit Symbol
0	0	1	Tvl Short
0	1	0	Tv2 Long
0	1	1	Tv3 SOF or EOD
1	0	0	Tv6 IFS
1	0	1	Tv5 IBS
1	1	0	>Tv5 BREAK

A.3.3.2 SYMBOL DECODER—Similarly to the operation of the symbol encoder, the completion of the reception of a symbol from the IDR by the SED causes the SED to interrupt the control microcomputer. The control microcomputer reads the SED's R0, R1, and R2 outputs to determine which symbol was received, as defined by Table A6.

The meaning of each received VPW symbol is dependent on whether the symbol level is dominant or passive. This level is read by the control microcomputer on the Receive Level RECLEV Output of the SED.

TABLE A6—RECEIVE SYMBOL DEFINITION

Outputs R2	Outputs R1	Outputs R0	Input Description J1850 Rec. Symbol
0	0	0	Invalid Symbol
0	0	1	Tv1 Short
0	1	0	Tv2 Long
0	1	1	Tv3 SOF or EOD
1	0	0	Tv4 EOF
1	0	1	Tv6 IFS
1	1	0	Tv5 IBS

A.3.3.3 INVALID SYMBOL—The reception of an Invalid Symbol (Refer to Table A6) sets a Noise Flag NF Output. This Output remains at a logic "1" level until cleared by Reset Noise Flag RSTNF Input.

A.3.3.4 SED INPUTS AND OUTPUTS—All inputs and outputs from the SED are at standard CMOS voltage levels.

The following discusses some particular SED inputs and outputs.

A.3.3.4.1 Transmit Clear (TRCLR~) Input—The SED's TRCLR~ Input can be used by the control microcomputer to terminate the transmission of an IBS or Break symbol. IBS may be terminated after the minimum Tv5 period has been generated and a Break symbol may be terminated after a period greater than the maximum Tv5 has been generated.

A.3.3.4.2 Reset (RST) Inputs—The RST Input to the SED is used to initialize the SED.

A.3.3.4.3 Oscillator (OSC) Input—The OSC Input to the SED is the Time Base Clock Signal. When the DS Input is connected to VCC, a 4.0 MHz clock is required. When DS Input is grounded, an 8.0 MHz clock is required.

A.3.3.4.4 Transmit Strobe (TRSTRB) Input—The TRSTRB Input to the SED operates as a watchdog circuit to insure that the control microcomputer is functioning correctly. The TRSTRB Circuit requires that each transmit symbol must be strobed into the SED or the SED will transmit an idle Output. The TRSTRB Input is active when ENTRSTRB Input is grounded and disabled when the ENTRSTRB Input is connected to VCC.

A.3.4 Integrated Driver/Receiver (IDR)—The IDR is the interface between the vehicle bus wiring and the SED. The IDR integrates and simultaneously performs both the bus driver and receiver functions. The bus output waveform is specifically shaped to conform to SAE J1850 requirements and to minimize EMI.

A.3.4.1 BUS OUTPUT WAVESHAPING—The IDR bus output is wave shaped to have symmetrical rise and fall voltage waveforms within the range of allowed bus loading tolerances and battery operating voltages and during message arbitration. The waveshaping consists of a set rise/fall and corner rounding times. The maximum rise/fall time is 16 µs with the corner rounding of 4 µs. The typical rise/fall time is set to 14 µs with corner rounding of 3 µs.

A.3.4.2 TRANSMITTING SIGNALS ON THE BUS—The IDR receives 1's and 0's from the SED on the CMOS compatible Transmit pin (Tx) and translates these to wave shaped high and low signals at the bus out pin (Bout). The propagation delay between the transition on the Tx pin and the corresponding transition on the Bout pin measured at 3.875 V is a maximum of 12 µs, with a typical value of 10 µs.

A.3.4.3 PROCESSING SIGNALS RECEIVED FROM THE BUS—The IDR receives 1's and 0's on the bus input pin (Bin) and translates these to 1's and 0's at the CMOS compatible Receive pin (Rx). The propagation delay between the transition on the Rx pin and the corresponding transition on the Bin pin measured at 3.875 V is a maximum of 2 μs, with a typical value of 1 μs.

A.3.5 Additional Capabilities—The additional capabilities of the HBCC are shown in A.2.6.

The PCI is able to buffer both transmit and receive messages, send and receive Break, support both CRC and Checksum for message error checking, and comprehend IBS.

The MPU used in this example for the control microcomputer is a Motorola 68HC05C4. This particular version of the 68HC05 family is used for the designer's convenience. A much simpler 68HC05P7 microcomputer could also be used to perform the required PCI control tasks and in some cases the host tasks of a scan tool. The 68HC05C4's utilization during message transmission averages about 50% of the available CPU time. It is likely that with so much CPU time still available, the 68HC05C4 or the more powerful and faster 68HC11 could be used to perform both the control and host tasks.

A second example in this section Figure A3 shows the use of the control microprocessor associated with the PCI as the host also. As such the control microcomputer is connected to the HBCC and ISO 9141-2 interfaces. The external equipment manufacturer is left to design the interface to any required keyboard and display, which are not shown. Similarly to the case shown in Figure A2A, the control microcomputer externally selects the HBCC or the ISO 9141-2 interface when needed.

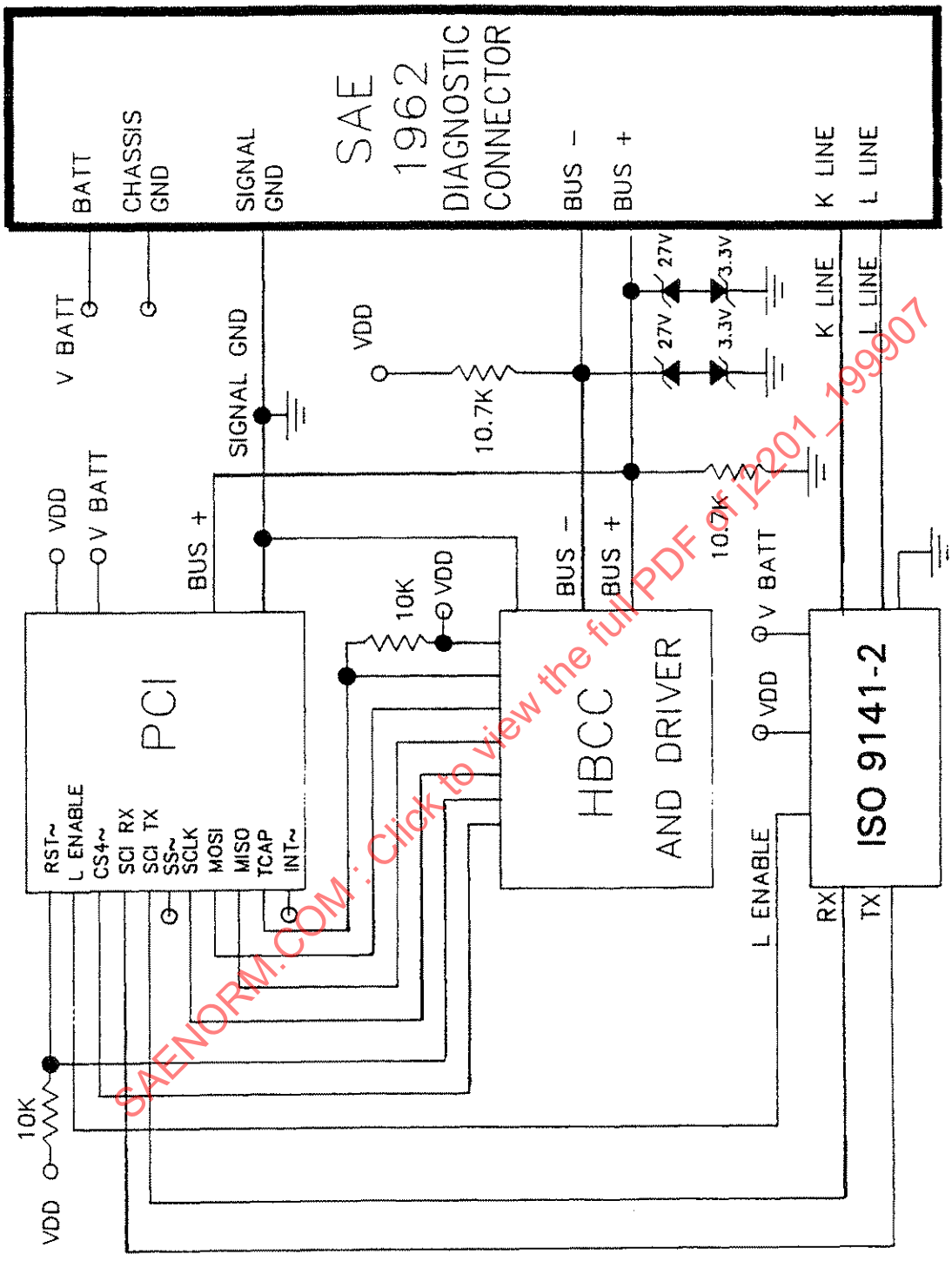


FIGURE A3—PCI, HBCC, AND ISO 9141-2 WITH PCI AS HOST ALSO

A.4 SGC-Thomson Protocol Engine Interface Example—This example (see Figure A4a) combines the use of an SGS-Thomson ST9 microprocessor as a control microcomputer, control microprocessor software, a GAL6001 programmable logic array device and discrete bus transceiver circuity to provide the required interface support. Together these devices are identified as the SGS-Thomson Protocol Engine. (Figure A4B shows the detail of the ST9 and the GAL6001, Figure A4C shows the detail of the 10.4 Kbps Transceiver, Figure A4D shows the detail of the 41.6 Kbps Transceiver, and Figure A4E shows the detail of the ISO 9141 Transceiver.)

In this example the host microcomputer is connected to only one interface, the Protocol Engine. The Protocol Engine directly supports the required SAE J1850 and ISO 9141-2 interfaces and includes the ability to, on its own, determine the particular interface being used by a vehicle to support OBD II communication.

A.4.1 General Overview—As mentioned in Section A.4, the Protocol Engine is comprised of four parts: the ST9 control microcomputer, control microcomputer software, the GAL6001 programmable logic array device, and several discrete transceivers.

A.4.2 Control Microcomputer—A SGS-Thomson ST9 family microcomputer is used as an interface control microcomputer. The ST9 microcomputer is a general purpose 8/16 bit microcomputer. Its timer facilities are used extensively to provide the control of transmitted waveforms and the detection of received signals.

Both SPI and parallel interfaces to host microcomputers are supported.

The host can request the Protocol Engine to either scan the possible OBD II interfaces and determine the type used in a given vehicle or use a selected interface type.

A.4.2.1 CONTROL MICROCOMPUTER SOFTWARE STRUCTURE—The highest priority tasks in the control microcomputer support the symbol-by-symbol processing of each transmitted and received message frame. The control microcomputer, and in particular the ST9's timers, directly controls the rise and fall of transmitted signals and directly follows the rise and fall of received signals. When transmitting a message, it performs both of these processes simultaneously. All other control microcomputer tasks are processed on a time available basis.

The following further identifies the control microcomputer's tasks:

a. Highest Priority Tasks:

1. Bus synchronization
2. Transmission and reception of frame symbols
3. Bit-by-bit arbitration
4. Comprehension of IBS

b. Background tasks:

1. CRC or checksum calculation
2. Symbol encoding and decoding
3. Message buffering
4. In-frame response processing
5. Error checking
6. Host interface

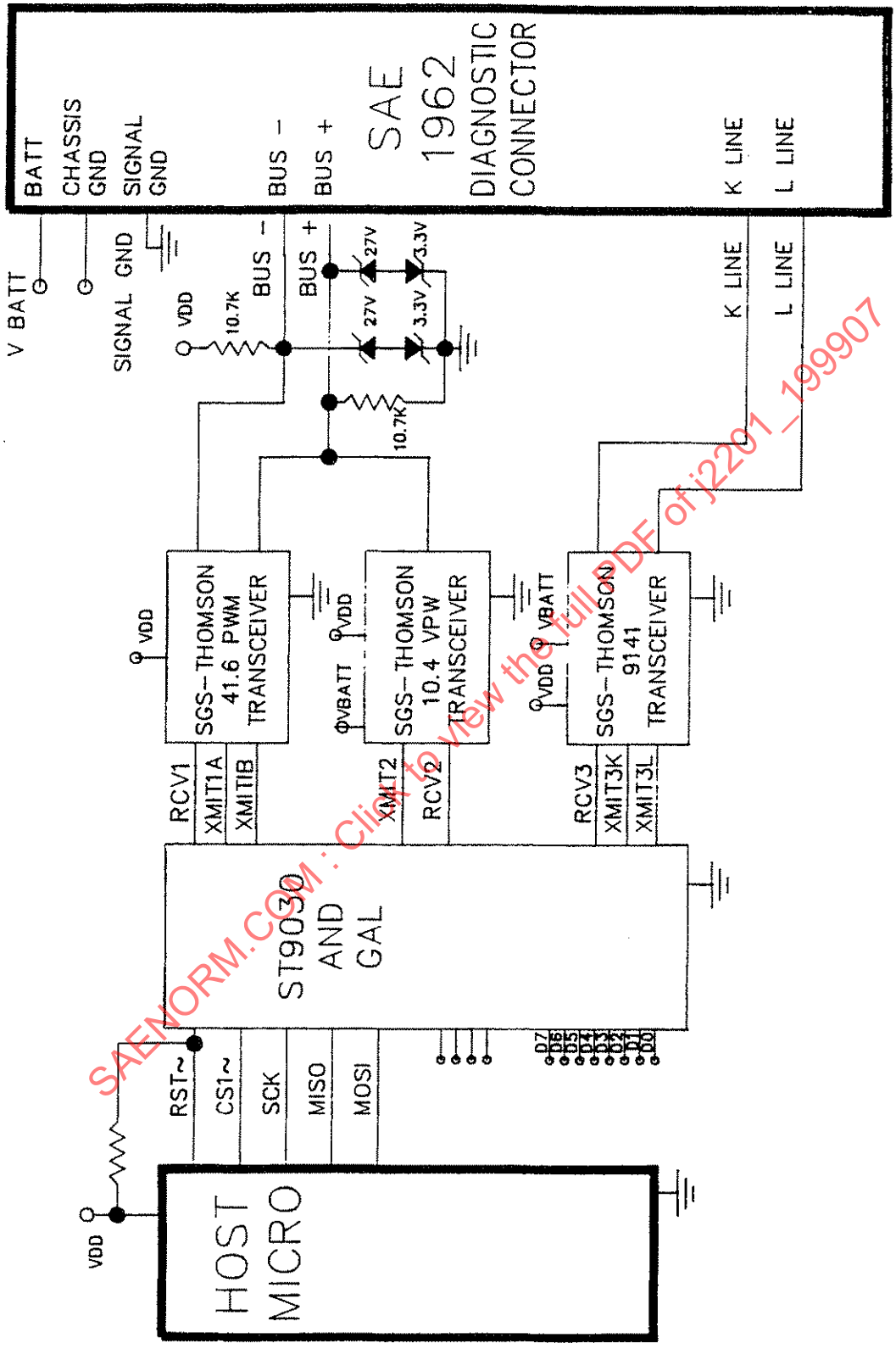


FIGURE A4A—SGS-THOMSON PROTOCOL ENGINE

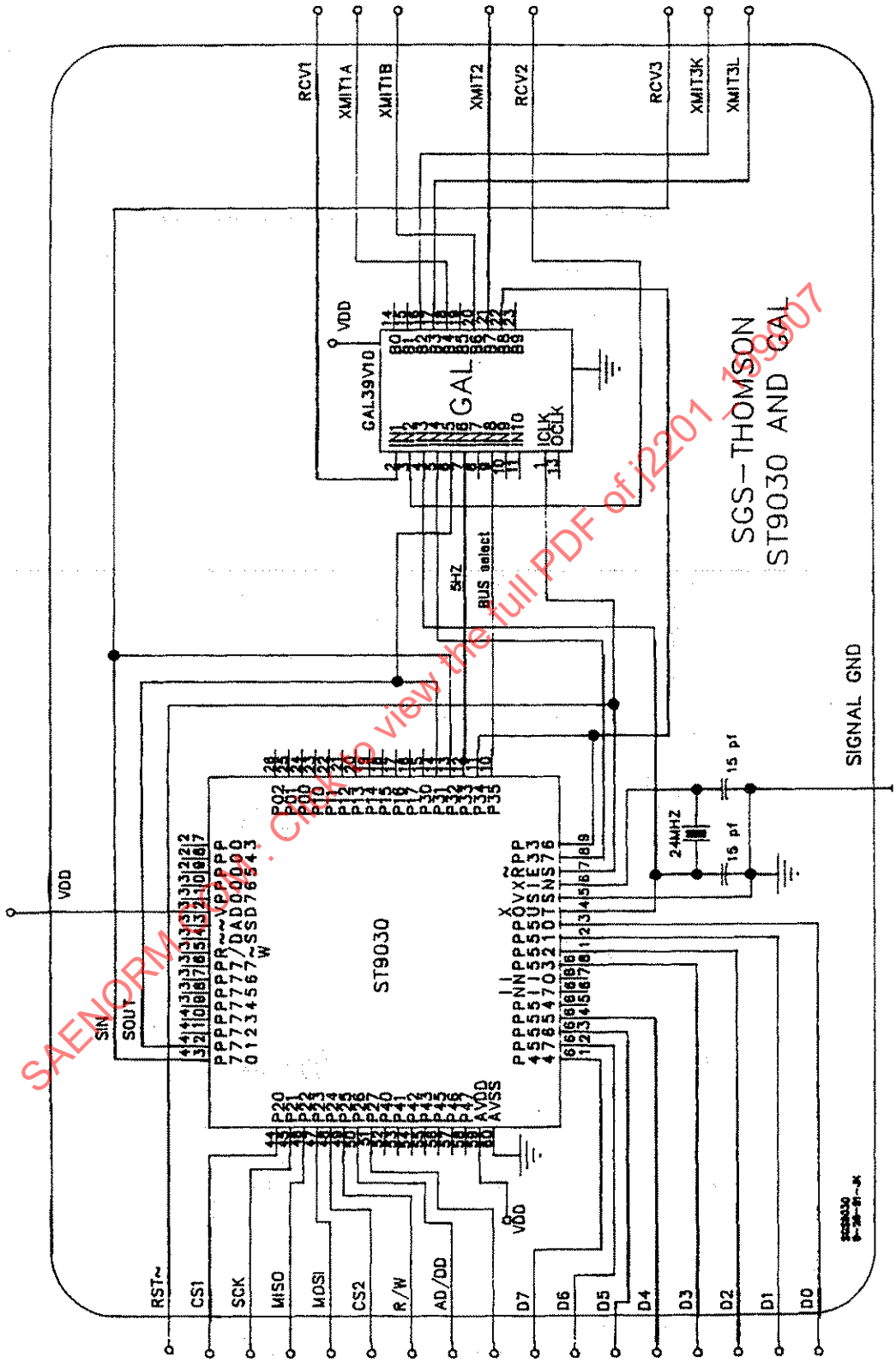


FIGURE A4B—SGS-THOMSON ST9030 AND GAL

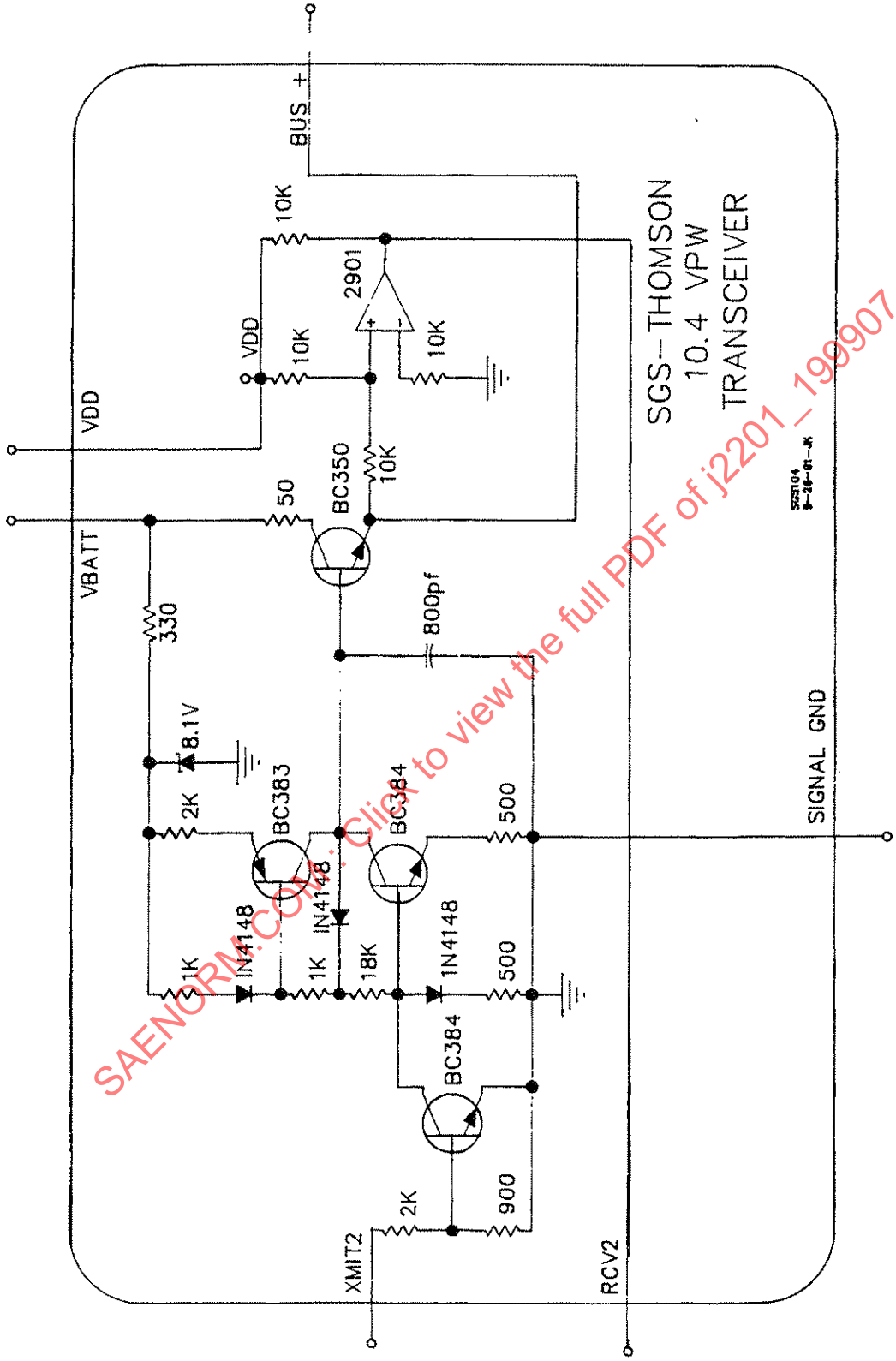


FIGURE A4C—SGS-THOMSON 10.4 VPW TRANSCEIVER

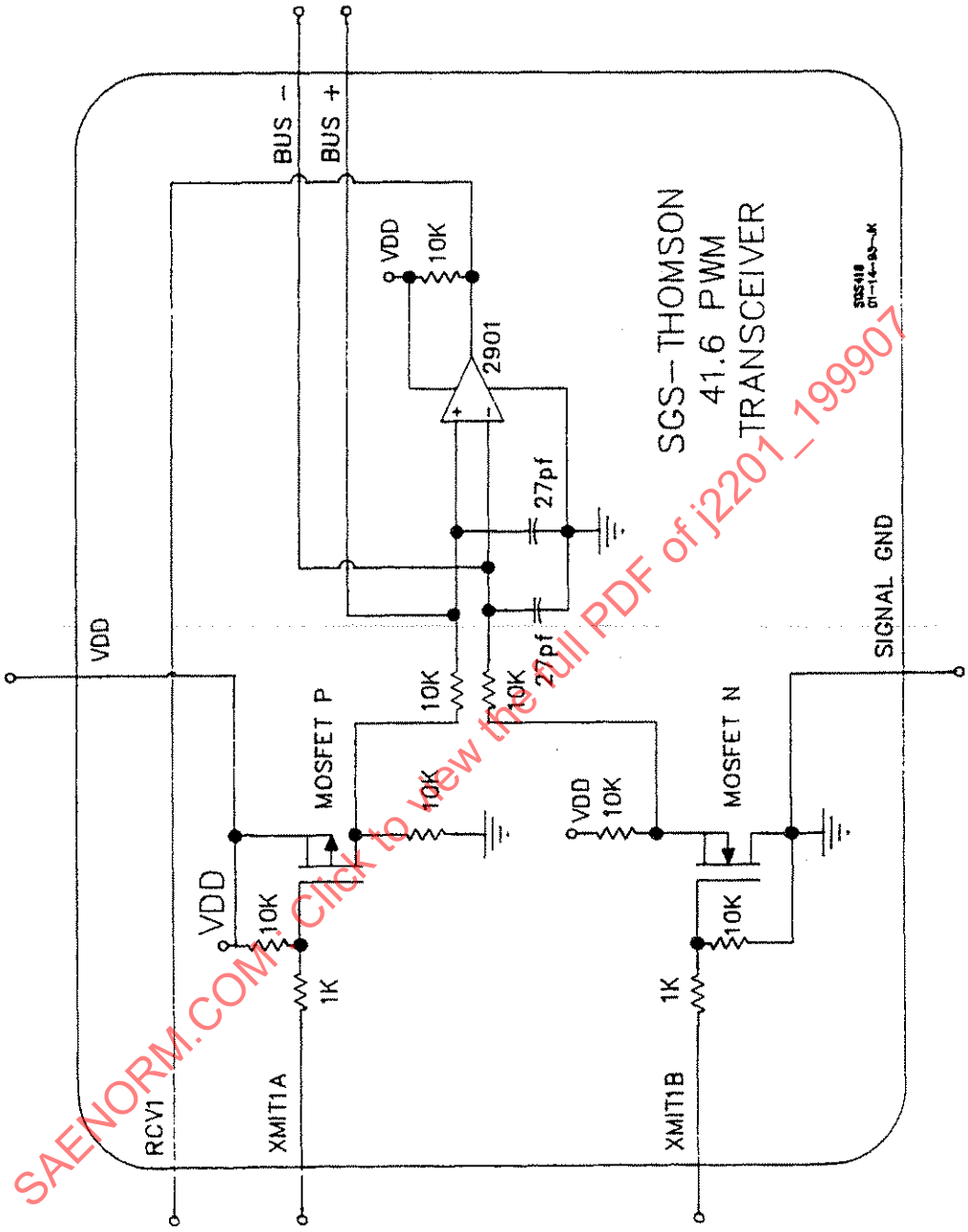


FIGURE A4D—SGS-THOMSON 41.6 PWM TRANSCEIVER

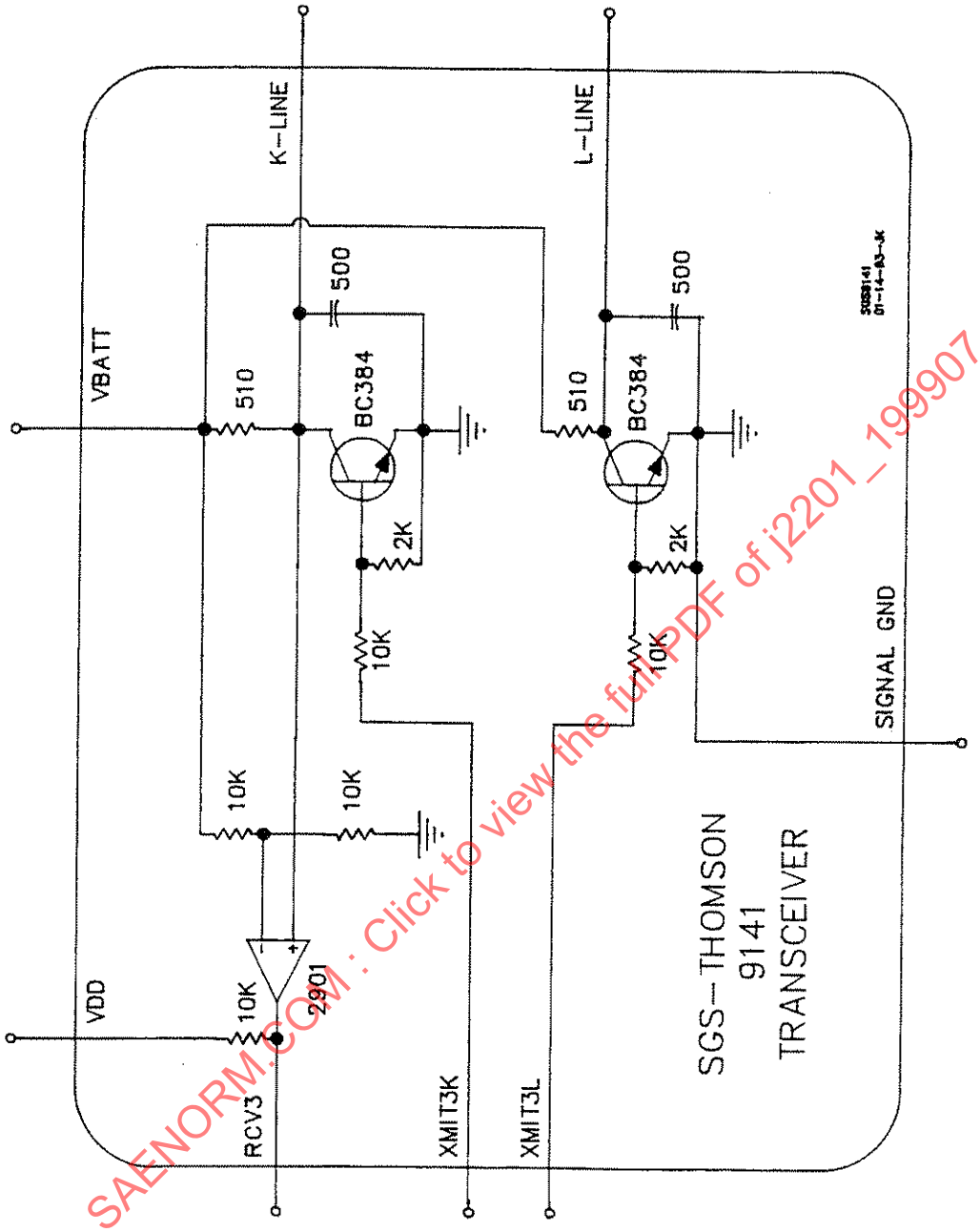


FIGURE A4E—SGS-THOMSON 9141 TRANSCEIVER

A.4.2.2 CONTROL MICROCOMPUTER INTERFACE TO THE HOST—Communication between the control microcomputer and the host is based on the host reading from and writing to 64 bytes of memory in the control microcomputer. Sixteen bytes are reserved for status and control of SAE J1850 related operations, 13 bytes are reserved for status and control of ISO 9141-2 related operations, 3 bytes are reserved for general status and control, and 32 bytes are reserved for receive and transmit buffers and related pointers and counters.

Data transfers between the host and the control microcomputer are performed as 2 byte transfers, a control byte and a data byte, initiated by the host. The control byte from the host indicates whether the host wishes to read or write a byte in the control microcomputer and what the address of the byte is Figure A5. The second SPI byte transfer moves the byte read from the control microcomputer to the host or the byte to be written from the host to the control microcomputer.

The control microprocessor supports both an SPI and a parallel type of interface with the host. In order to properly service bus communications, the control microcomputer controls the actual data transfers between itself and the host.

A.4.2.3 AUTOMATIC OBD II INTERFACE SCAN—Upon host request, the Protocol Engine will, by itself, automatically scan all of the possible OBD II interfaces and determine the one supported by a given vehicle. The interface scan sequence used is as follows:

a. SAE J1850 41.6 Kbps PWM
b. SAE J1850 10.4 Kbps VPW
c. ISO 9141-2

Bit	7	6	5	4	3	2	1	0
	R/W	R*	MSB	6 bits of address				LSB

R/W = Read/Write command bit. 0 = write to ST9
1 = read from ST9

R* = Reset. 0 = Resets the chip.

6 bits of address = address to read/write.

FIGURE A5—PROTOCOL ENGINE COMMAND BYTE

A.4.3 GAL6001 Programmable Logic Array—The GAL6001 Programmable Logic Array performs primarily as a digital filter of the signal received from the vehicle bus. It is a standard high-performance field programmable EEPROM logic device and is available in a variety of packages. The device chosen has a 20-year data retention specification. It is organized as a 78 x 64 x 36 FPLA.

A.4.4 Discrete Transceiver—The discrete transceiver circuity includes all the components necessary to create the required transmit signal waveforms on a bus and to receive bus signals. This includes support for 41.6 PWM, 10.4 VPW and ISO 9141-2. The transceiver is responsible for or controls the following:

a. Voltage on the bus
b. Current levels
c. Media impedance
d. Signal rise times and fall times

The overall signal transmitted by the transceiver is controlled by the control microcomputer. Signal transmission on a bus is created through the use of current sources/sinks. The 10.4 Kbps VPW driver is comprised of a current source of 1X magnitude and a current sink of 2X magnitude. By switching on and off the 2X current sink, the transceiver effectively causes the bus to make transitions from zero to one to zero. The current sinked or sourced into the bus capacitance controls the rise and fall times of the signals. The control microcomputer controls the "ON" and "OFF" timing of the 2X sink, thereby controlling the data and data rate on the bus.

By using the controlled current sink and source approach this driver effectively becomes short-circuit proof.

A.4.5 Additional Capabilities—In addition to supporting the requirements of this document, the Protocol Engine also supports the following options of SAE J1850: Checksum, comprehend IBS, Single Byte Header, and Break.

Being software driven, the control microcomputer can be considered for supporting additional custom interfaces to test equipment host microcomputers.

Simulation has determined that the bus signal control and monitor tasks require less than 30% of the control microcomputer leaving at least 70% of a very fast microcomputer available for the other tasks such as performing as the host.

Other facilities that are available with various versions of ST9 microcomputers are:

a. Eight 8-bit analog-to-digital channels
b. Eight user readable input pins
c. Eight user writable output pins

The analog to digital support would allow the host to have the control microcomputer to do such things as determine if the diagnostic connector is plugged in, and look at the voltages on the bus and determine if communication is possible.

ST9 microcomputer models have support for up 32K of ROM or up to 16 megabytes of external memory.

The ST9030 shown in this example is the lowest member in the ST9 family currently available, therefore all members of the ST9 family can support OBD II communication.

A.5 Silicon Systems F690/F691 Scan Tool Chipset Interface Example—This example (see Figure A6A) combines the use of the F690 Codec and F691 Transceiver scan tool chip set developed by Silicon Systems (SSi) to provide the required interface support. (Figure A6B shows the detail of the F690 Codec, and Figure A6C shows the details of the F691 transceiver.)

A.5.1 General Overview—The SSi Scan Tool chipset consists of the F690 Codec and the F691 Transceiver. The F690 Codec performs the bit/symbol level CODer/DECoder operations for the SAE J1850 interfaces. This allows the host processor to communicate with the chipset at the byte level. A parallel type interface is used between the host and the Codec.

The F691 Transceiver performs the physical interface for the SAE J1850 41.6 Kbps PWM, SAE J1850 10.4 Kbps VPW, and ISO 9141-2 requirements.

The ISO 9141-2 interface support of the Transceiver is directly connected to the SCI interface of the host.

A.5.2 F690 Codec—A functional block diagram of the F690 Codec chip is shown in Figure A6B. The Codec interfaces with the host processor via an 8-bit parallel bus. The bus is multiplexed and both Motorola and Intel microcomputer parallel type interfaces are supported. Reset, chip select, and interrupt line connections are also used.

Transmit bytes are loaded from the host into the Transmit FIFO buffer of the Codec, which also performs a parallel to serial conversion. The bit serial output from the Transmit FIFO is input into the symbol encoder as well as into the CRC generation logic. The symbol encoder converts bits (1 or 0) into the appropriate PWM/VPW timing symbols. Transmitted symbols are synchronized to the received data. The transmit logic controls the output that drives the F691 Transceiver.

The framing symbols for SOF, IBS, EOD, NB, EOF, and IFS are also generated by the Codec. The host only needs to pass to the Codec the message bytes, and checksum, when used.

The F690 Codec receives information from the F691 Transceiver chip via the RCVJ input pin. The receive input from the Transceiver is, in the Codec, digitally filtered and fed into the symbol decoder. Transitions at the digital filter output are also used to synchronize the transmitter. The collision detect logic compares the transmitted and received signals and performs the bus arbitration function.

The symbol decoder compares the digital filter output with known symbol templates. When a data bit match occurs, the proper bit (1 or 0) is shifted into the receive FIFO buffer. When framing symbols are detected, they are stripped from the received data and fed into the synchronization/controller logic. The data values from the symbol decoder output are also fed into the CRC checking logic. The receive FIFO buffer also performs a serial to parallel conversion such that the host processor can parallel read a received data byte.

Configuration data, transmit and receive status data, and interrupt mask registers are included on the F690 Codec. Interrupts can be enabled to indicate transmit buffer empty, receive buffer full, CRC error, lost attribution, idle bus, fault detection, etc.

The F690 Codec chip is available in a 28 PLCC package. The part is fabricated in a CMOS technology, operates from a 5 V supply, and consumes less than 100 mW max.

A.5.3 F691 Transceiver—A functional block diagram of the F691 Transceiver is shown in Figure A6C. This chip performs the physical level interface functions. The Transceiver is capable of driving a single-ended, waveshaped, 10.4 Kbps VPW signal as well as the differential, 41.6 Kbps PWM signal onto SAE J1850 buses. Two configuration lines from the F690 Codec determine what xmit/receive mode is to be used by the Transceiver. The receiver consists of either single-ended or differential comparators. Hysteresis is added to improve noise immunity. The transmitter/receiver for the ISO 9141-2 interface is also included on the Transceiver.

Other than waveshaping for the SAE J1850 10.4 Kbps VPW interface, all timing is controlled by the F690 Codec chip. An on-chip bandgap reference and prepackage trims are used to tightly control the SAE J1850 10.4 Kbps VPW waveshape parameters and comparator switching levels.

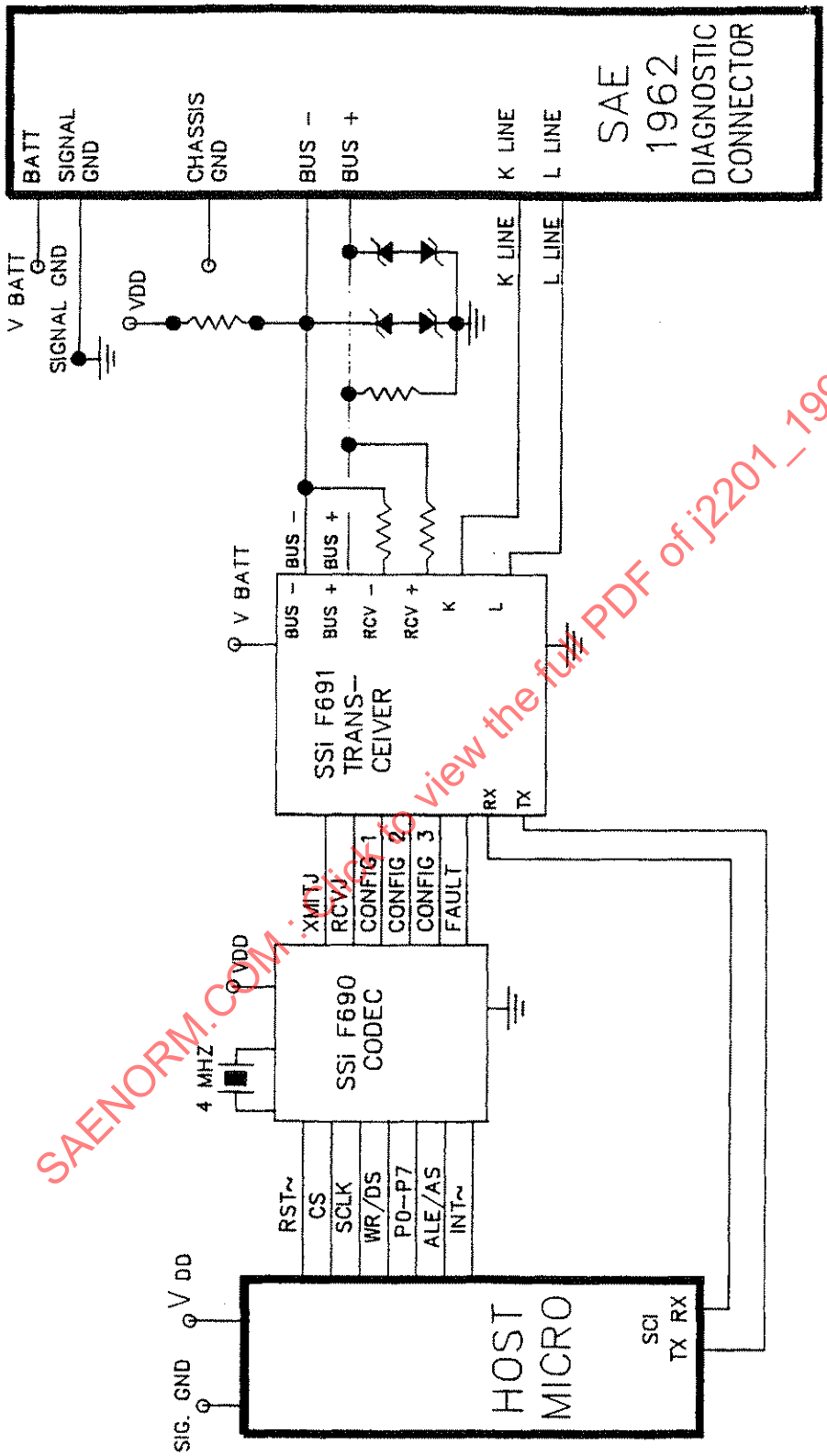


FIGURE A6A—SILICON SYSTEMS F690/F691 CHIPSET

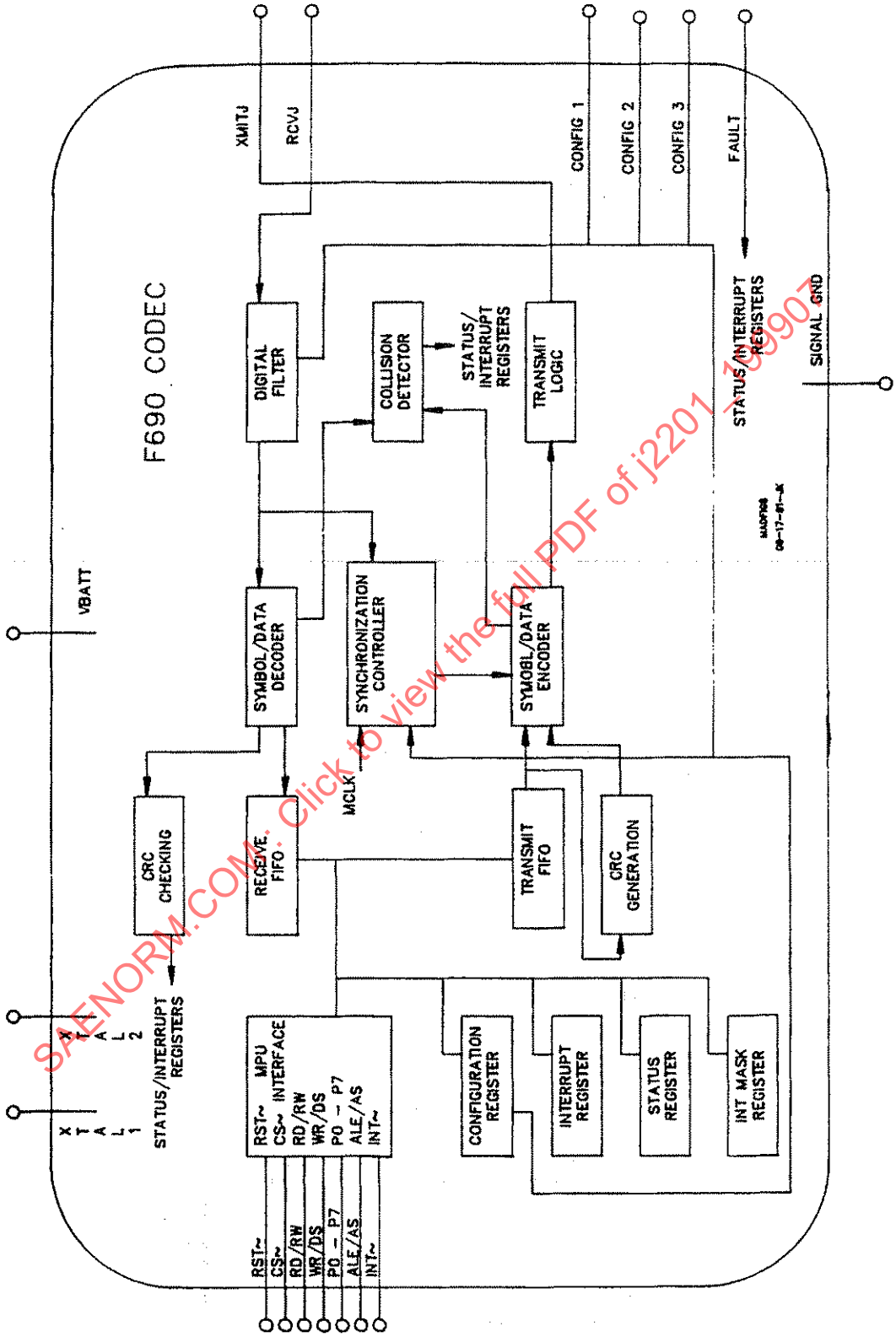


FIGURE A6B—F690 CODEC

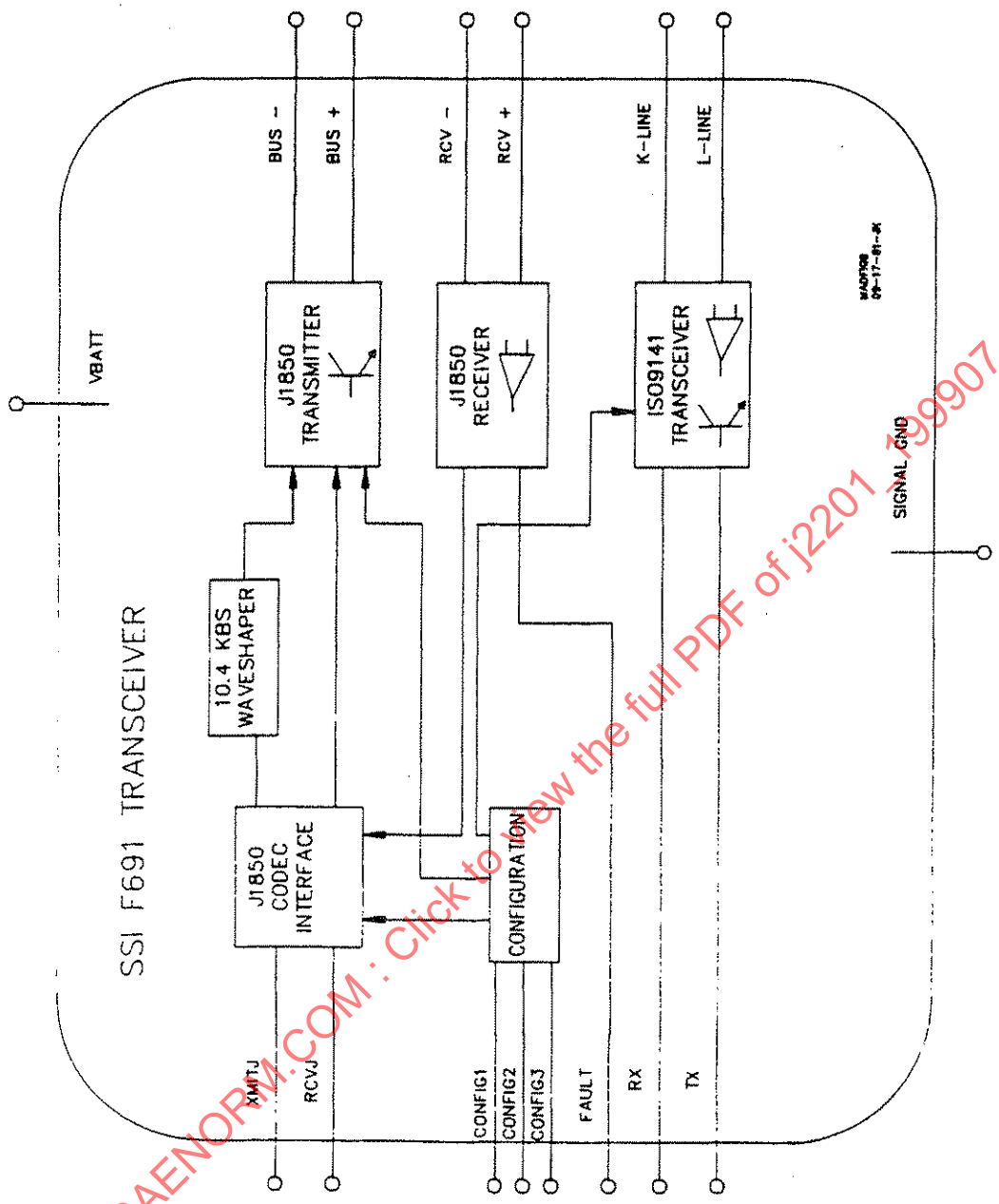


FIGURE A6C—F691 TRANSCEIVER

The F691 Transceiver chip is available in a 16-pin plastic DIP. The part is fabricated in a BIPOLAR technology, operates from a single 9 to 13 V supply, and dissipates 500 mW max.