MC68HC58

DATA LINK CONTROLLER

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part.

MCUinit, MCUasm, MCUdebug, and RTEK are trademarks of Motorola, Inc. Motorola and the Motorola logo are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

How to reach us:
USA/EUROPE/Locations Not Listed: Motorola Literature Distribution;
P.O. Box 20912; Phoenix, Arizona 85036. 1-800-441-2447 or 602/303-5454
MFAX: RMFAX0@email.sps.mot.com - TOUCHTONE (602) 244-6609
INTERNET: http://Design-NET.com
JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC,
6F Seibu-Butsuryu-Center, 3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 81-3-3521-8315
ASIA PACIFIC: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,
51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298

TABLE OF CONTENTS

Paragraph **Title** **Page**

SECTION 1INTRODUCTION

SECTION 2SIGNAL AND PIN DESCRIPTIONS

TABLE OF CONTENTS
(Continued)

Paragraph **Title** **Page**

SECTION 3 J1850 FRAME FORMAT

TABLE OF CONTENTS
(Continued)

Paragraph **Title** **Page**

SECTION 4 DATA LINK CONTROLLER OPERATION

SECTION 5 CONTROL AND STATUS CODES

TABLE OF CONTENTS
(Continued)

Paragraph **Title** **Page**

TABLE OF CONTENTS
(Continued)

Paragraph **Title** **Page**

APPENDIX A ELECTRICAL CHARACTERISTICS

APPENDIX B MECHANICAL DATA AND ORDERING INFORMATION

APPENDIX C DLC REGISTERS

TABLE OF CONTENTS
(Continued)

Paragraph Title Page

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(Continued)

Figure **Title** **Page**

LIST OF TABLES

Table **Title** **Page**

LIST OF TABLES
(Continued)

SECTION 1INTRODUCTION

The MC68HC58 DLC (data link controller) handles microcontroller unit (MCU) to Society of Automotive Engineers (SAE) J1850 bus interface duties. The MC68HC58 DLC is the successor to the MC68HC56 DLCP (data link controller parallel) and the MC68HC57 DLCS (data link controller serial). The MC68HC58 is pin configurable to communicate with a host MCU via an 8-bit non-multiplexed parallel data bus or a Motorola serial peripheral interface.

The DLC consists of control logic and bus transceiver circuits. **Figure 1-1** shows the internal structure of a DLC configured for parallel mode. **Figure 1-2** shows the internal structure of a DLC configured for serial mode. The built-in bus transceiver allows the DLC to be directly connected to the J1850 bus, thus providing a complete link between the central processing unit (CPU) host application and the J1850 bus. The J1850 bus protocol is a method of information transfer via messages (frames) between nodes. A node is any location on the J1850 bus that sends and receives messages.

The following are primary features of the DLC:

- SAE J1850 compatible
- Class 2 (vehicle bus communication protocol) compatible
- Handles all network protocol functions (access, arbitration, error detection)
- Supports polled or interrupt host DLC servicing
- Message buffering on transmit and receive
- On-board transceiver with waveshaping
- Operates with up to a 2-volt ground offset between network nodes
- Pin configurable SPI or parallel host interface
- Digitally filtered receiver
- Host configurable oscillator divisor
- Power conserving sleep feature with fast wakeup on bus or host activity
- High voltage CMOS (40 volt HVCMOS) process
- Built-in transient and ESD protection

The DLC handles SAE J1850 frames with minimal MCU servicing. Each DLC can be operated in either interrupt mode or polled mode. Internal first in/first out (FIFO) buffers, 20 bytes for receiver data and 11 bytes for transmitter data, allow full frame length operations. The MCU typically transfers complete frames to the DLC for transmission on the SAE J1850 bus, and is interrupted only when a complete frame is received from the SAE J1850 bus. The DLC handles all arbitration, error detection, and optional in-frame response duties internally.

Changes to the operating configuration can be made at any time. Depending upon the command, the changes can be made immediately or following the current J1850 bus transaction.

The logic section of the DLC consists of the MCU interface, transmit and receive bit timing logic, transmit and receive FIFO buffer logic, the control and timing block (framing, error detection and bus arbitration), and command and status control logic.

The bus transceiver allows the DLC to be directly connected to the J1850 bus, thus providing a complete link between the MCU application and the J1850 bus.

Transceiver operation is constrained by available power and the need to function reliably in the presence of conducted and induced noise. The main source of conducted noise is the ground offset between nodes. The DLC operates correctly under any combination of offsets up to a maximum differential of two volts at any frequency. Induced noise tends to be composed of short-duration pulses. The receive bit timing section includes a digital filter to remove these pulses.

The transceiver provides a waveshaped seven volt serial analog signal in response to a timed signal from the bit timing logic. The transceiver also receives J1850 bus waveforms, and provides the control logic with unfiltered inputs.

To achieve the 7-volt signal level necessary for the J1850 bus, the transceiver has a separate 9- to 16-volt power supply input (V_{BATT}). The transceiver actively drives the J1850 bus high, and passively allows an RC load to pull the J1850 bus down. If ground is lost, the transceiver releases the J1850 bus. The transceiver also protects the MCU interface by not passing on any disruptive signals that may be on the J1850 bus.

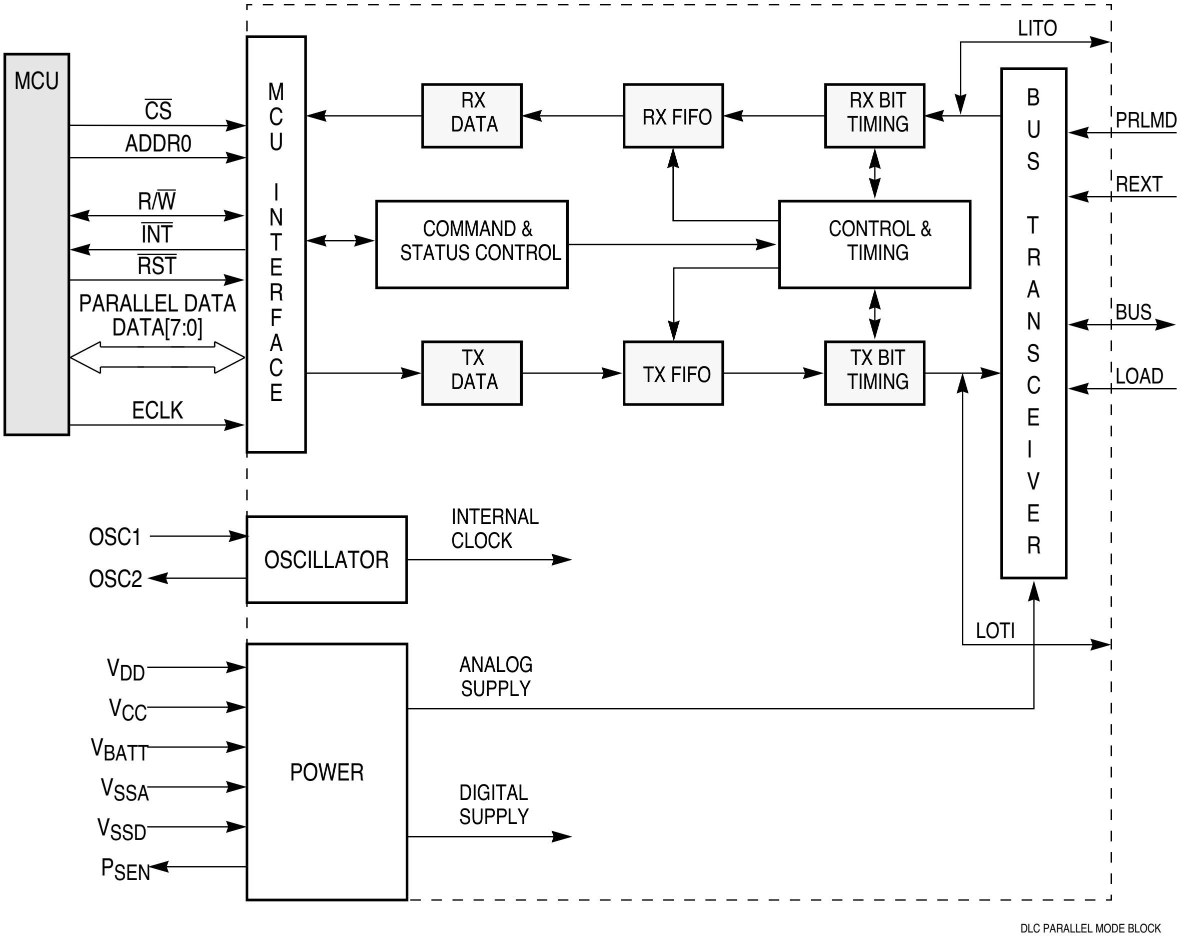


Figure 1-1 MC68HC58 DLC Parallel Mode Block Diagram

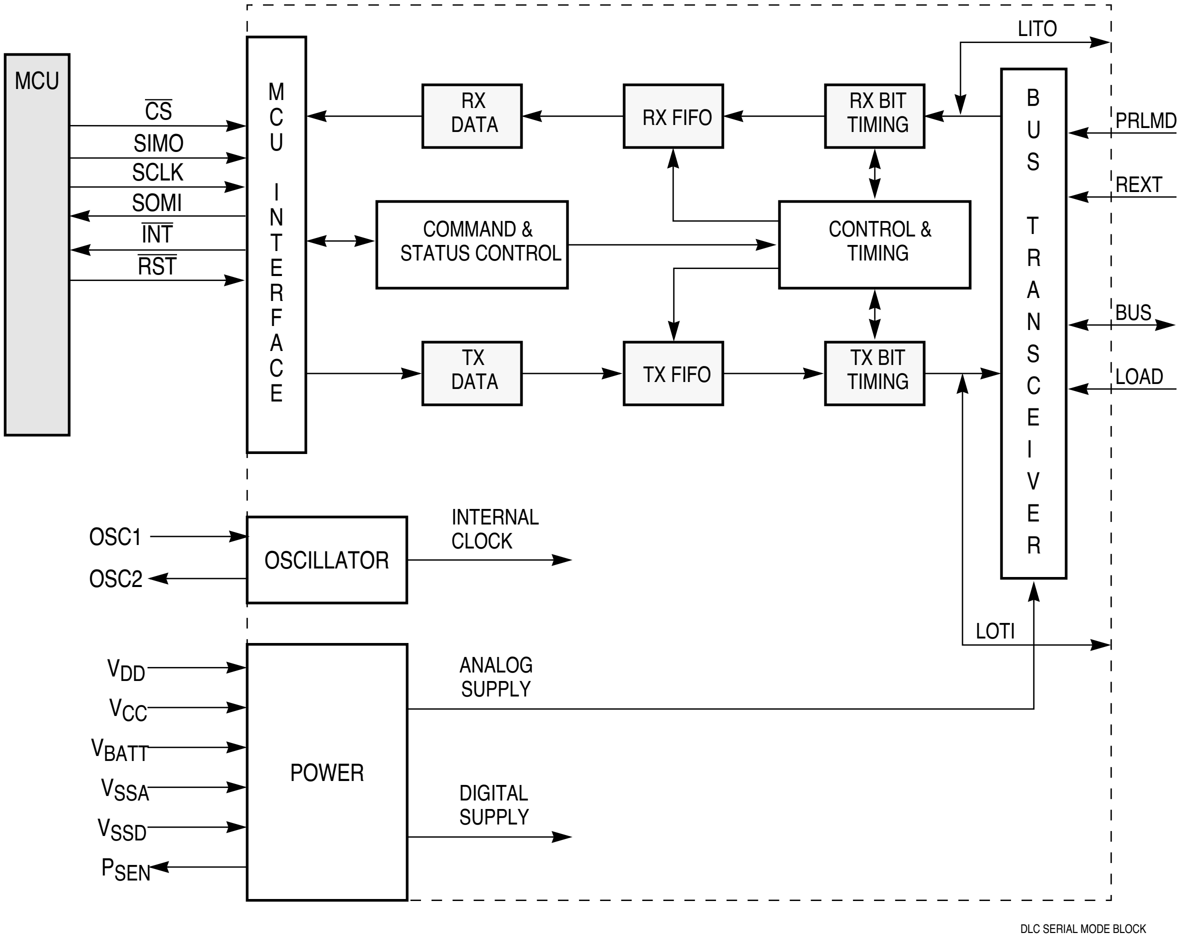


Figure 1-2 MC68HC58 DLC Serial Mode Block Diagram

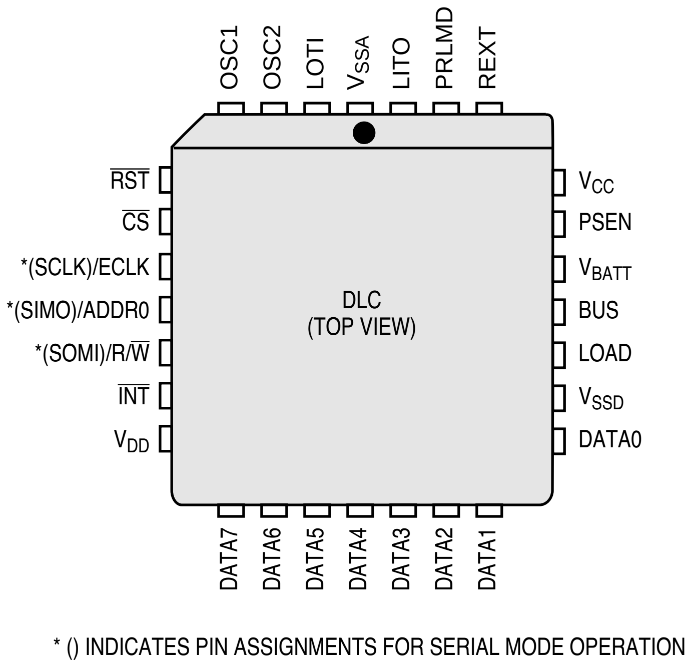
SECTION 2SIGNAL AND PIN DESCRIPTIONS

The MC68HC58 DLC is available in a 28-pin plastic leaded chip carrier (PLCC) package and a 28-pin small outline integrated circuit (SOIC) package. The MC68HC58 is pin configurable to communicate with a host MCU via a serial or parallel interface. Serial or parallel mode is selected by connecting the PRLMD pin to ground or V_{DD} respectively. Pin function and recommended interconnections are discussed in the following paragraphs. Refer to **APPENDIX B MECHANICAL DATA AND ORDERING INFORMATION** for information on package dimensions and ordering information.

2.1 MC68HC58 DLC Parallel Mode

When the PRLMD pin is tied to V_{DD}, the MC68HC58 DLC communicates with the host MCU via a parallel interface. Interface timing is based on the M6800 external bus clock signal (ECLK). Motorola M68HC11, M68HC16, and M68300 series MCUs have ECLK outputs, but any host MCU that meets the timing specification can be used.

Figure 2-1 is a pinout of the MC68HC58 (PLCC).



DLC PIN ASSIGNMENT

Figure 2-1 MC68HC58 DLC Pin Assignments

2.1.1 DLC Parallel Mode Pin Function

Table 2-1 summarizes DLC pin functions when operating in parallel mode. Detailed discussion of each function follows. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for more information on electrical specifications.

Table 2-1 MC68HC58 DLC Parallel Mode Pin Function

Name	Type	Function
ADDR0	Input	Address select signal
BUS	Input/Output	Serial data signal
$\overline{\text{CS}}$	Input	DLC chip-select signal
DATA[7:0]	Input/Output	Bidirectional three-state data bus
ECLK	Input	6800 bus clock
$\overline{\text{INT}}$	Output	DLC interrupt request
LITO	Input/Output	Logic in transceiver out
LOAD	Input	External bus load connection
LOTI	Input/Output	Logic out transceiver in
OSC1	Input	External clock connection
OSC2	Output	External reference connection
PSEN	Output	Power supply status signal
PRLMD	Input	Parallel/serial mode select signal
REXT	Input	Transceiver biasing resistor
$\overline{\text{RST}}$	Input	DLC reset signal
R/$\overline{\text{W}}$	Input	DLC data transfer control
V_{BATT}	Power supply	Transceiver power connection
V_{CC}	Power supply	Analog power connection
V_{DD}	Power supply	Digital power connection
V_{SSA}	Power supply	Analog ground
V_{SSD}	Power supply	Digital ground

2.1.1.1 ADDR0 — Address Bit

This pin is used in conjunction with the R/$\overline{\text{W}}$ signal to address the DLC in the MCU memory map. Although the name ADDR0 implies that the pin should be connected to address line 0, it can in fact be connected to any address line to place it at a desired location in memory.

2.1.1.2 BUS — SAE J1850 Multiplex Bus

This pin connects the DLC to the SAE J1850 multiplex bus. The bus signal is driven to a nominal 7 Vdc with respect to the bus load when in an active level; it is grounded through the bus load when in a passive level.

2.1.1.3 $\overline{\text{CS}}$ — DLC Chip-Select

This pin is used to input the DLC parallel data exchange enable signal. It has a nominal 15 kΩ internal pull-up resistor.

2.1.1.4 DATA[7:0] — DLC Data Bus

These pins are the bidirectional data lines used to transfer parallel bytes to and from the DLC. The data lines are in high-impedance state unless $\overline{\text{CS}}$ is asserted.

2.1.1.5 ECLK — 6800 Bus Clock

This pin is used to input a 6800 peripheral bus clock. Clock timing controls parallel data exchange with the MCU. M68HC11, M68HC16, and M68300 MCUs all provide clock outputs to facilitate connection of 6800-timed peripherals.

2.1.1.6 $\overline{\text{INT}}$ — DLC Interrupt Request

This pin is used to output an open drain active-low interrupt request signal to the MCU. The signal is fully compatible with M68HC11, M68HC16, and M68300 interrupt request inputs. $\overline{\text{INT}}$ must have an external pull-up resistor.

2.1.1.7 LITO — Logic In Transceiver Out

This pin is an external tap on the internal receive signal sent from the transceiver to the control logic. It is used for testing only, and must be left unconnected for normal operation.

2.1.1.8 LOAD — External Bus Load

This pin provides an internal ground connection for the SAE J1850 multiplex bus load resistor. If DLC analog ground is lost, LOAD goes to a high-impedance state.

2.1.1.9 LOTI — Logic Out Transceiver In

This pin is an external tap on the internal transmit signal sent from the control logic to the transceiver. It is used for testing only, and must be left unconnected for normal operation.

2.1.1.10 OSC1, OSC2 — External Oscillator

The DLC can operate with either an external clock signal or an external ceramic resonator. These pins support connection of both types of clock. If an external clock source is used, the signal is input via OSC1, and OSC2 is left floating. If a resonator is used, it is connected between OSC1 and OSC2.

2.1.1.11 PRLMD — Parallel Mode

This pin is used to select the desired MCU interface mode. When this pin is pulled up to V_{DD}, the parallel mode is selected.

2.1.1.12 PSEN — Power Supply Enable

This pin provides an external V_{BATT} source when the DLC is active. It assumes high-impedance state when the DLC is in standby mode.

2.1.1.13 $\overline{\text{RST}}$ — DLC Reset

This pin is used to input an active-low system reset signal. $\overline{\text{RST}}$ must have an external pull-up resistor.

2.1.1.14 REXT — External Bias Resistor

This pin connects an external bias resistor to the transceiver. The resistor value determines the waveform of the transmitted bus signal.

2.1.1.15 R/$\overline{\text{W}}$ — Read/Write Strobe

This pin is used to input a data direction control signal to the DLC. This signal is used in conjunction with the ADDR0 signal to select the DLC at a specific address in the MCU memory map.

2.1.1.16 V_{BATT} — Battery Voltage

This pin connects a separate switched or unswitched 12 Vdc power supply to the DLC bus transceiver. This supply should be well regulated and protected against switching transients.

2.1.1.17 V_{CC} — Analog Power Supply Voltage

This pin connects a nominal 5 Vdc power supply to the analog transceiver circuitry in the DLC. For maximum noise immunity, V_{CC} supply path should be separate from V_{DD} supply path.

2.1.1.18 V_{DD} — Digital Power Supply Voltage

This pin connects a nominal 5 Vdc power supply to the digital control circuitry in the DLC. For maximum noise immunity, the V_{DD} supply path should be separate from V_{CC} supply path.

2.1.1.19 V_{SSA} — Analog Power Ground

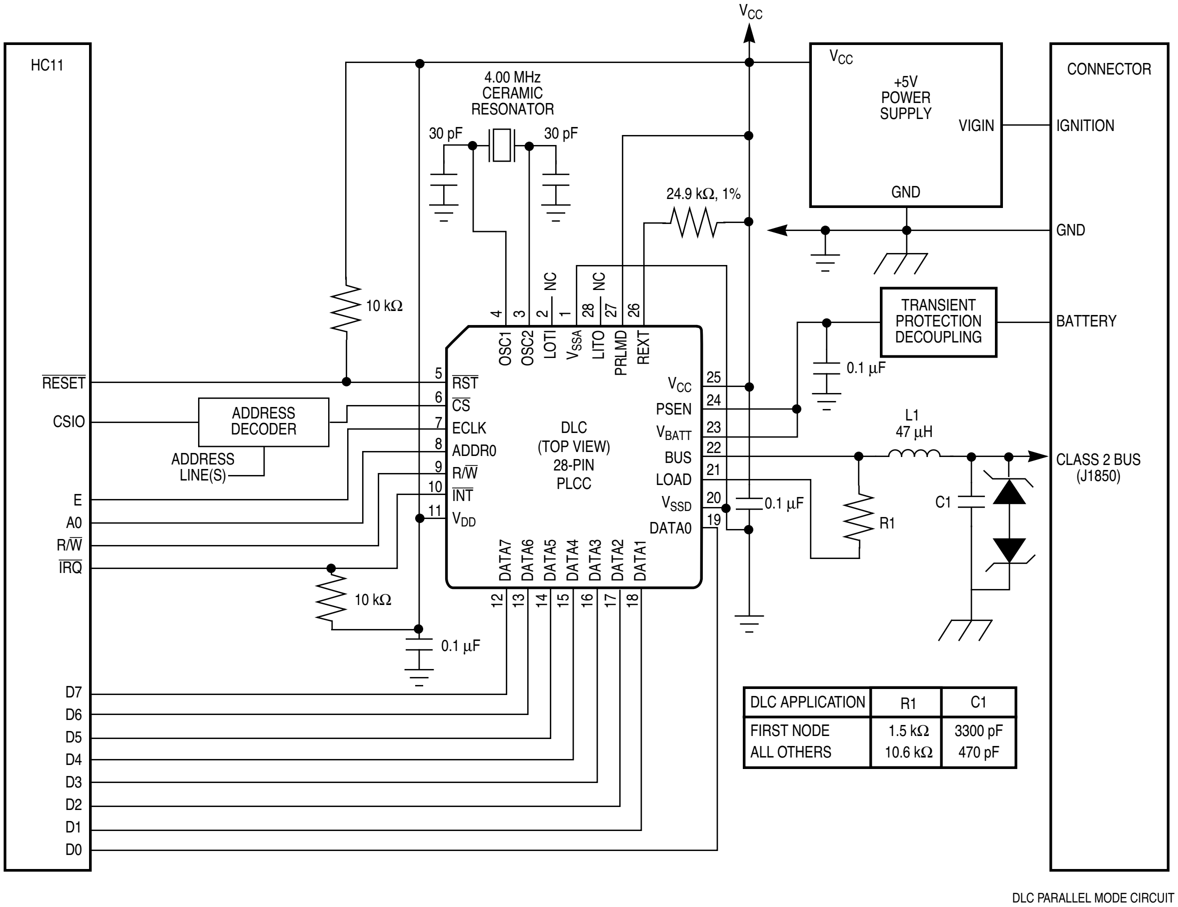
This pin provides the analog power ground connection to the DLC. Loss of analog ground directly affects operation of the LOAD pin. For best noise immunity in operation, the V_{SSA} ground path should be separate from the V_{SSD} ground path.

2.1.1.20 V_{SSD} — Digital Power Ground

This pin provides the digital power ground connection to the DLC. For best noise immunity in operation, the V_{SSD} ground path should be separate from the V_{SSA} ground path.

2.1.2 Example DLC Parallel Mode System

Figure 2-2 shows a typical DLC circuit (PLCC). The component values displayed are recommended, although adjustment may be required in actual operation. The example is shown wired for low-power standby mode. Typical M68HC11 MCU connections are shown, but any host MCU that has appropriate inputs and outputs compatible with DLC signals can be used.



DLC APPLICATION	R1	C1
FIRST NODE	1.5 kΩ	3300 pF
ALL OTHERS	10.6 kΩ	470 pF

Figure 2-2 DLC Parallel Mode Circuit

The following additional guidelines apply to **Figure 2-2**:

1. Pull-up resistor values depend on electrical characteristics of the host MCU.
2. Exact values for the external components are a function of printed circuit board (PCB) capacitance and inductance, socket capacitance, operating voltage and crystal technology.
3. The example ceramic resonator is a Murata CSA4.00MGA with typical load capacitance. The DLC contains an internal 1 MΩ resistor across OSC1 and OSC2. No external resistor is required.
4. L1 is a surface mount inductor which should have characteristics similar to TDK NL322522T-470J-3.
5. The external bias resistor (REXT) determines the waveshape of J1850 bus signals transmitted by the DLC.
6. For maximum noise immunity, V_{CC} and V_{DD} should be supplied by separate lines; V_{SSA} and V_{SSD} should also be separate. The V_{BATT} pin is shown attached to the permanent “unswitched” battery supply to take advantage of the better transient protection found on this circuit rather than the “switched” battery, or ignition.
7. Applications whose requirements for electro-static discharge (ESD) protection exceed the level provided by the bus pin internal circuitry and the bus loading components may require additional transient protection. The example in **Figure 2-2** illustrates this by including two 16 volt zener diodes placed between the bus and ground. These diodes (part # P4SMA16AT3) should be located as close to the module connector as possible. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for information on maximum voltage ratings.
8. One J1850 node in the vehicle typically has a heavier BUS to LOAD pin loading. This allows a smaller differential between the total network load of two nodes versus 32 nodes. If 26 nodes or more are used, the heavier loading should not be present as this may violate the maximum capacitance and minimum resistance allowed by the J1850. A network of 26 or more nodes should all have the lower load values.
9. **Figure 2-2** reflects a configuration for no wake-up upon the detection of J1850 bus activity. If system wake-up upon the detection of J1850 bus activity is desired, perform the following:
    - Tie the V_{DD} pin to a 5 Vdc power supply. This 5 Vdc power supply is powered down when the DLC is placed in the standby mode.
    - Tie the PSEN pin on the DLC through a 33 kΩ resistor to the ignition input of the power supply/regulator. This limits the current sourced by the DLC. This resistor value should be determined by the individual application. A 10 kΩ pull-down resistor should be included from the PSEN line to ground. A 100 kΩ resistor from the $\overline{\text{RST}}$ pin to ground should also be added.
    - Tie the V_{CC} pin and the pull-up points for the external bias resistor (REXT) on the DLC to a separate 5 Vdc power supply. This 5 Vdc power supply remains powered up when the rest of the node is powered down.

2.2 MC68HC58 DLC Serial Mode

The MC68HC58 DLC communicates with the host MCU via a standard serial peripheral interface (SPI) when in serial mode. Interface timing is based on a serial clock signal (SCLK) provided by the host MCU. Motorola M68HC05, M68HC11, M68HC16, and M68300 series MCUs have SPI capabilities, but any host MCU that meets the clock specification can be used. Refer to **Figure 2-1** for serial mode pinout information.

2.2.1 DLC Serial Mode Pin Function

Table 2-2 summarizes DLC serial mode pin function. Detailed discussion of each function follows. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for more information on electrical specifications.

Table 2-2 MC68HC58 DLC Serial Mode Pin Functions

Name	Type	Function
BUS	Input/Output	Serial bus connection
$\overline{\text{CS}}$	Input	DLC chip-select signal
$\overline{\text{INT}}$	Output	DLC interrupt request
LITO	Input/Output	Logic in transceiver out signal
LOAD	Input	External bus load connection
LOTI	Input/Output	Logic out transceiver in signal
OSC1	Input	External clock connection
OSC2	Output	External reference connection
PSEN	Output	Power supply status signal
PRLMD	Input	Parallel/serial mode select signal
REXT	Input	Transceiver biasing resistor
$\overline{\text{RST}}$	Input	DLC reset signal
SCLK	Input	SPI serial clock
SIMO	Input	Slave in master out signal
SOMI	Output	Slave out master in signal
V_{BATT}	Power supply	Transceiver power connection
V_{CC}	Power supply	Analog power connection
V_{DD}	Power supply	Digital power connection
V_{SSA}	Power supply	Analog ground
V_{SSD}	Power supply	Digital ground

2.2.1.1 BUS — SAE J1850 Multiplex Bus

This pin connects the DLC to the SAE J1850 multiplex bus. The bus signal is driven to a nominal 7 Vdc with respect to the bus load in an active level; it is grounded through the bus load in a passive level.

2.2.1.2 $\overline{\text{CS}}$ — DLC Chip-Select

This pin is used to input the DLC serial data exchange enable signal. It has a nominal 15 kΩ internal pull-up resistor.

2.2.1.3 $\overline{\text{INT}}$ — DLC Interrupt Request

This pin is used to output an open drain active-low interrupt request signal to the MCU. The signal is fully compatible with M68HC05, M68HC11, M68HC16, and M68300 interrupt request inputs. $\overline{\text{INT}}$ must have an external pull-up resistor.

2.2.1.4 LITO — Logic In Transceiver Out

This pin is an external tap on the internal receive signal sent from the transceiver to the control logic. It is used for testing only, and must be left unconnected for normal operation.

2.2.1.5 LOAD — External Bus Load

This pin provides an internal ground connection for the SAE J1850 multiplex bus load resistor. If DLC analog ground is lost, LOAD goes to a high-impedance state.

2.2.1.6 LOTI — Logic Out Transceiver In

This pin is an external tap on the internal transmit signal sent from the control logic to the transceiver. It is used for testing only, and must be left unconnected for normal operation.

2.2.1.7 OSC1, OSC2 — External Oscillator

The DLC can operate with either an external clock signal or an external ceramic resonator. These pins support connection of both types of clock. If an external clock source is used, the signal is input via OSC1, and OSC2 is left floating. If a resonator is used, it is connected between OSC1 and OSC2.

2.2.1.8 PRLMD —Parallel Mode

This pin is used to select the desired MCU interface mode. When this pin is pulled down from V_{DD}, the serial mode is selected.

2.2.1.9 PSEN — Power Supply Enable

This pin provides an external V_{BATT} source when the DLC is active. It assumes high-impedance state when the DLC is in standby mode.

2.2.1.10 $\overline{\text{RST}}$ — DLC Reset

This pin is used to input an active-low system reset signal. $\overline{\text{RST}}$ must have an external pull-up resistor.

2.2.1.11 REXT — External Bias Resistor

This pin connects an external bias resistor to the transceiver. The resistor value determines the waveform of the transmitted BUS signal.

2.2.1.12 SCLK — Serial Clock

This pin is used to input a serial clock signal to the DLC SPI. This signal is generated by the SPI bus master device (typically the MCU).

2.2.1.13 SIMO — Slave In Master Out

The SPI interface performs simultaneous bidirectional transfers initiated by a bus master. This pin connects the serial data input from the SPI bus to the DLC (the DLC is an SPI slave device).

2.2.1.14 SOMI — Slave Out Master In

The SPI performs simultaneous bidirectional transfers initiated by a bus master. This pin connects the serial data output from the DLC to the SPI bus (the DLC is an SPI slave device).

2.2.1.15 V_{BATT} — Battery Voltage

This pin connects a separate switched or unswitched 12 Vdc power supply to the DLC bus transceiver. This supply should be well regulated and protected against switching transients.

2.2.1.16 V_{CC} — Analog Power Supply Voltage

This pin connects a nominal 5 Vdc power supply to the analog transceiver circuitry in the DLC. For maximum noise immunity, V_{CC} supply path should be separate from V_{DD} supply path.

2.2.1.17 V_{DD} — Digital Power Supply Voltage

This pin connects a nominal 5 Vdc power supply to the digital control circuitry in the DLC. For maximum noise immunity, the V_{DD} supply path should be separate from V_{CC} supply path.

2.2.1.18 V_{SSA} — Analog Power Ground

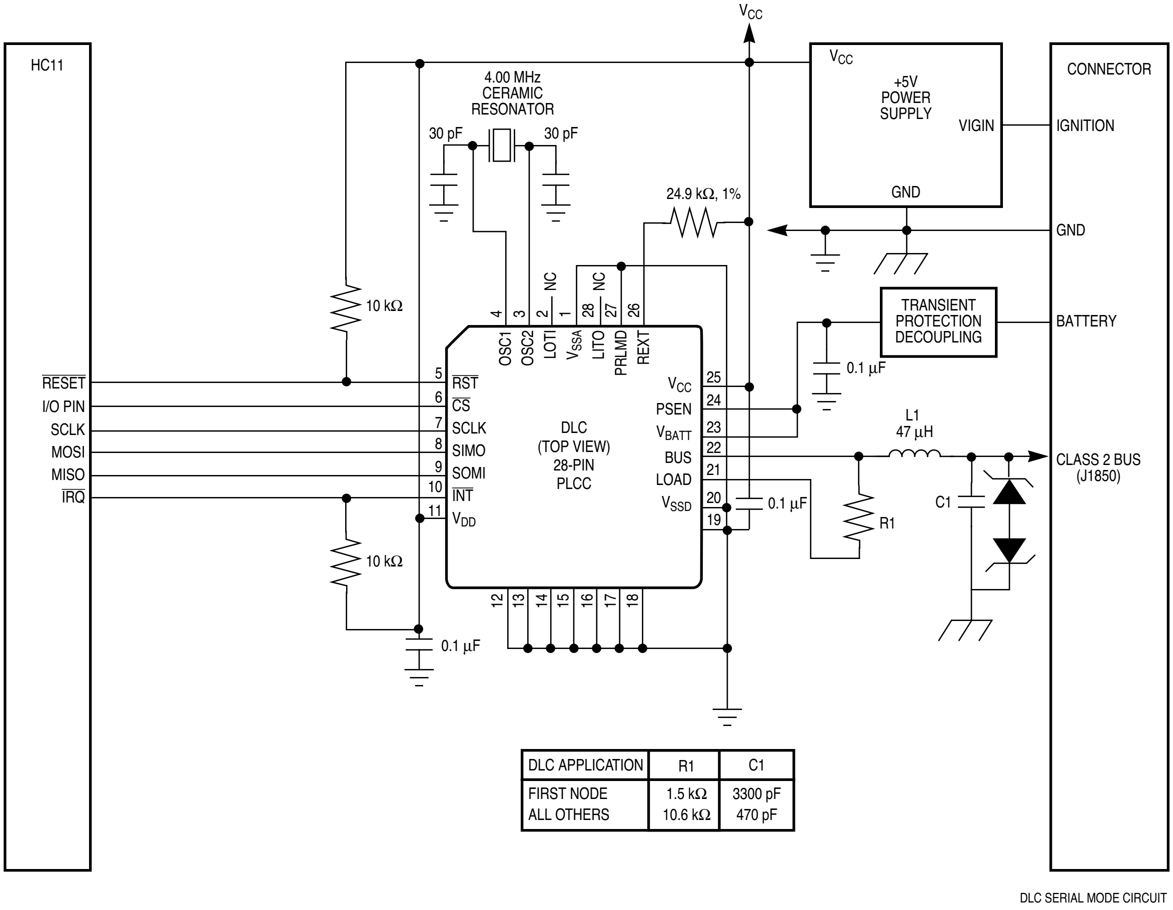
This pin provides the analog power ground connection to the DLC. Loss of analog ground directly affects operation of the LOAD pin. For best noise immunity in operation, the V_{SSA} ground path should be separate from the V_{SSD} ground path.

2.2.1.19 V_{SSD} — Digital Power Ground

This pin provides the digital power ground connection to the DLC. For best noise immunity in operation, the V_{SSD} ground path should be separate from the V_{SSA} ground path.

2.2.2 Example DLC Serial Mode System

Figure 2-3 displays a typical DLC serial mode circuit (PLCC). The component values displayed are recommended, although adjustment may be required in actual operation. The example is shown wired for low-power standby mode. Typical M68HC11 MCU connections are shown, but any host MCU that has appropriate inputs and outputs compatible with DLC signals can be used.



DLC APPLICATION	R1	C1
FIRST NODE	1.5 kΩ	3300 pF
ALL OTHERS	10.6 kΩ	470 pF

Figure 2-3 DLC Serial Mode Circuit

The following additional guidelines apply to **Figure 2-3**:

1. Pull-up resistor values depend on electrical characteristics of the host MCU.
2. Exact values for the external components are a function of printed circuit board (PCB) capacitance and inductance, socket capacitance, operating voltage and crystal technology.
3. The example ceramic resonator is a Murata CSA4.00MGA with typical load capacitance. The DLC contains an internal 1 MΩ resistor across OSC1 and OSC2. No external resistor is required.
4. L1 is a surface mount inductor which should have characteristics similar to TDK NL322522T-470J-3.
5. An external bias resistor (REXT) determines the waveshape of J1850 bus signals transmitted by the DLC.
6. For maximum noise immunity, V_{CC} and V_{DD} should be supplied by separate lines; V_{SSA} and V_{SSD} should also be separate. The V_{BATT} pin is shown attached to the permanent “unswitched” battery supply to take advantage of the better transient protection found on this circuit rather than the “switched” battery, or ignition.
7. Applications whose requirements for electro-static discharge (ESD) protection exceed the level provided by the bus pin internal circuitry and the bus loading components may require additional transient protection. The example in **Figure 2-3** illustrates this by including two 16-volt zener diodes placed between the bus and ground. These diodes (part # P4SMA16AT3) should be located as close to the module connector as possible. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for information on maximum voltage ratings.
8. One J1850 node in the vehicle typically has a heavier BUS to LOAD pin loading. This allows a smaller differential between the total network load of two nodes versus 32 nodes. If 26 nodes or more are used, the heavier loading should not be present as this may violate the maximum capacitance and minimum resistance allowed by the J1850. A network of 26 or more nodes should all have the lower load values.
9. Although pins 12-19 of the DLC are used only for factory testing, it is recommended that these pins be tied to ground in the application. This prevents additional current from being consumed by the DLC, particularly when the low-power standby mode is utilized.
10. **Figure 2-3** reflects a configuration for no wake-up upon the detection of J1850 bus activity. If system wake-up upon the detection of J1850 bus activity is desired, perform the following:
    - Tie the V_{DD} pin to a 5 Vdc power supply. This 5 Vdc power supply is powered down when the DLC is placed in the standby mode.
    - Tie the PSEN pin on the DLC through a 33 kΩ resistor to the ignition input of the power supply/regulator. This limits the current sourced by the DLC. This resistor value should be determined by the individual application. A 10 kΩ pull-down resistor should be included from the PSEN line to ground. A 100 kΩ resistor from the $\overline{\text{RST}}$ pin to ground should also be added.
    - Tie the V_{CC} pin and the pull-up points for the external bias resistor (REXT) on the DLC to a separate 5 Vdc power supply. This 5 Vdc power supply remains powered up when the rest of the node is powered down.

2.3 Bus Loading

The total load capacitance (C_{LOAD}) on the J1850 VPW network, as specified in SAE J1850, must be between 2470 pF and 16544 pF. Likewise, the load resistance (R_{LOAD}) on the network must be between 315 Ω and 1575 Ω. In addition, the network time constant (the product of R_{LOAD} and C_{LOAD}) must not exceed 5.2 µs.

It is recommended that in J1850 VPW systems with less than 26 nodes, one node should have a load capacitance of 3200 to 3300 pF and a load resistance of 1.5 kΩ, with all other nodes having the nominal load of 470 pF and 10.6 kΩ, as outlined in **SAE STANDARD J1850 – CLASS B DATA COMMUNICATIONS NETWORK INTERFACE**. This helps to minimize the differences in loading between small and large systems. When there are more than 26 nodes in a system, the large single load should be replaced with a nominal load to avoid exceeding maximum capacitance and minimum resistance specifications.

2.4 DLC Clock Sources

There are two types of clock signals associated with each DLC. The logic, or system clock, provides a reference frequency for internal operations and SAE J1850 bus operation. Host interface clocks provide timing for transfers between the DLC and the host MCU.

2.4.1 Logic Clock

The DLC can operate with an external oscillator reference connected between OSC1 and OSC2, or with an external clock source applied to OSC1 (OSC2 left floating). Internal clock frequency is determined by the value of the configuration byte oscillator divisor (OSCD) field. The DLC can be configured for divisor values of 1, 2, 3, or 4. Regardless of external clock frequency, in order to operate properly, a DLC must be configured so that its internal clock frequency is 2 MHz.

It is recommended that the DLC be clocked by a ceramic resonator. Ceramic resonators stabilize much more rapidly than crystal references (typically, 100 times faster), and are less expensive, although they may have looser frequency tolerance. The DLC can be configured to operate with 2, 4, 6, or 8 MHz resonators.

The DLC can operate with external clock input frequencies of 2, 4, 6, or 8 MHz. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for more information.

2.4.2 Host Interface Clock

Host interface circuitry within each type of DLC is clocked by a source in the host MCU. When operating in parallel mode, the DLC is clocked by an M6800 bus clock signal (CLK). In serial mode, the DLC is clocked by an SPI serial clock signal.

Either of the interface clocks can have any frequency from dc to 4.2 MHz. Duty cycle must be 50% ± 5%. The clocks need only be active during the time that the DLC $\overline{CS}$ signal is asserted. Refer to **SECTION 4 DATA LINK CONTROLLER OPERATION** for more information on host MCU clocking.

2.5 Power Supply Connections

During standby mode, the DLC oscillator and most internal circuitry are inactive. Power consumption is reduced significantly.

The host MCU must send the DLC a command byte to initiate low power operation, but the DLC can be reactivated in a number of ways. A DLC can come out of standby when it detects J1850 bus activity, when its status register is read by the host MCU, or when power is applied. If interrupts are enabled, the DLC can issue an interrupt request to the host MCU when it is activated.

The way in which power is applied to the DLC affects activation out of standby mode. Ignition switch position, battery drain during engine cranking, and alternator power stability can affect the supplies available at the time J1850 bus activity brings the DLC out of standby mode.

There are three recommended methods of implementing operation. Methods 1 and 2 permit the receiver to be activated by a frame on the J1850 bus. With a 4 or 8 MHz resonator, a properly configured DLC can accurately receive the frame that activates it. If interrupts are enabled, methods 1 and 3 permit the DLC to generate an interrupt request upon activation. Method 2 does not permit the DLC to generate an interrupt request when it is activated, because the control logic is not powered when J1850 bus activity is detected. Refer to **SECTION 4 DATA LINK CONTROLLER OPERATION** for more information on DLC interrupt service requests.

2.5.1 Method 1 — All Supplies Applied

When V_{BATT}, V_{CC}, and V_{DD} are applied, the PSEN pin goes to a high-impedance state when the DLC is in standby mode, and sources 12 Vdc when the DLC is activated. Figure 2-4 shows J1850 BUS, $\overline{\text{INT}}$, $\overline{\text{RST}}$, and PSEN timing relationships. If the host does not service the interrupt due to bus activity, the DLC automatically re-enters standby mode. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for more information on parameter specifications.

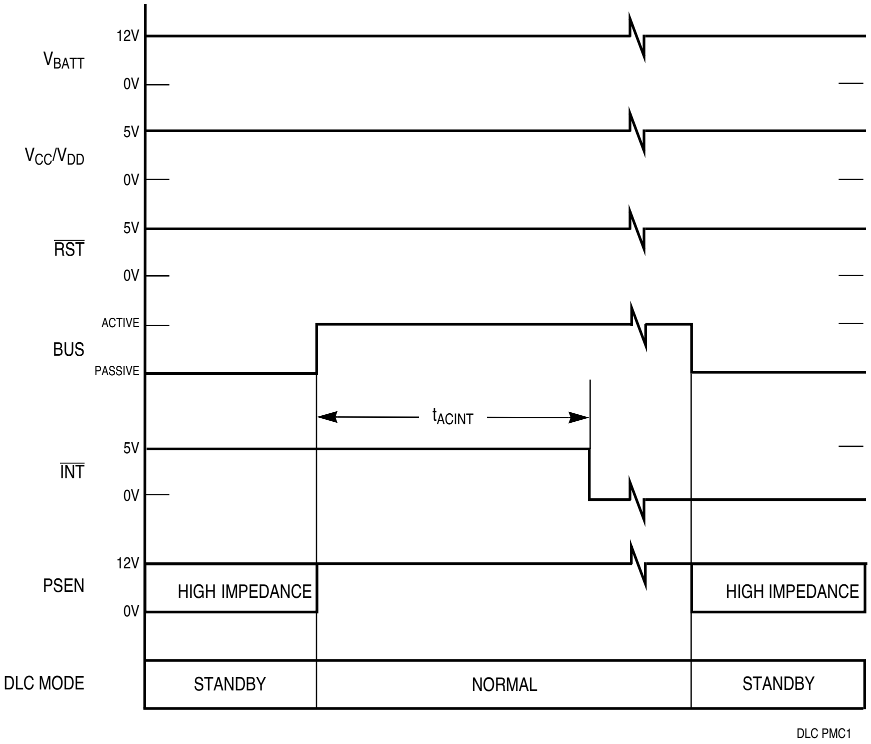


Figure 2-4 Method 1 — Standby Mode Supplies Applied

2.5.2 Method 2 — Switching V_{DD} With PSEN

With V_{BATT} and V_{CC} applied, the PSEN pin goes to a high-impedance state when the DLC is in standby mode and sources 12 Vdc when the DLC detects J1850 bus activity. Refer to **2.6 Reset** for more information on supply stabilization. **Figure 2-5** shows BUS, $\overline{\text{INT}}$, $\overline{\text{RST}}$, and PSEN timing relationships. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for more information on parameter specifications.

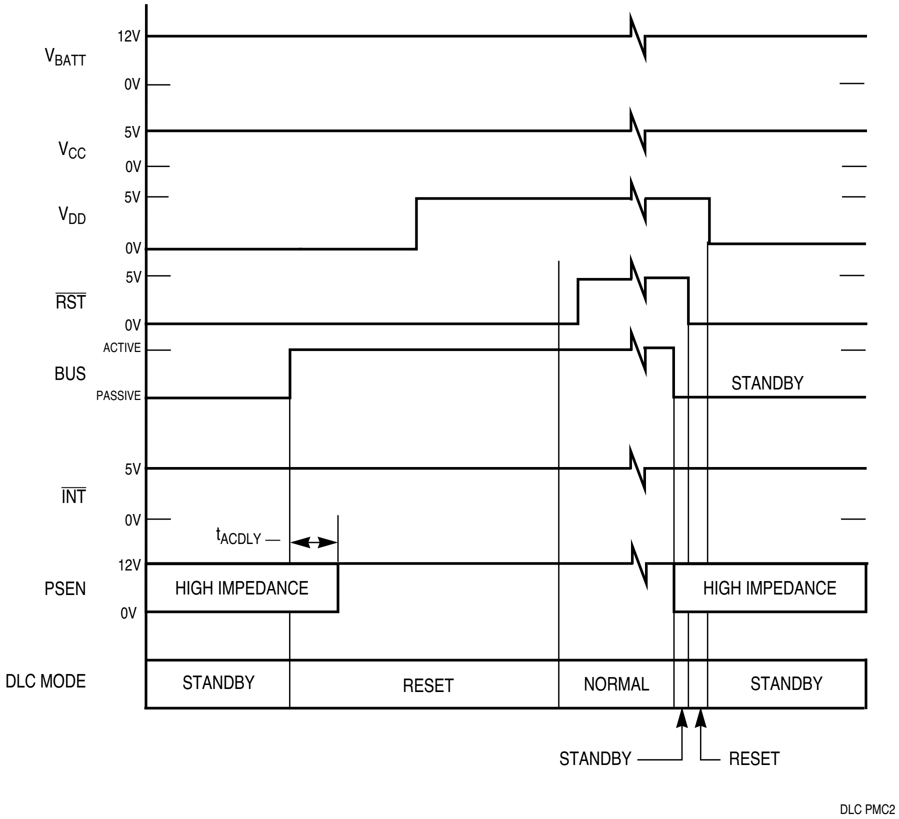


Figure 2-5 Method 2 — Switching V_{DD} with PSEN

2.5.3 Method 3 — Switching V_{DD} and V_{CC} Simultaneously

When V_{BATT} alone is applied, the PSEN pin stays in a high-impedance state until V_{DD} and V_{CC} are applied. Applying V_{DD} and V_{CC} simultaneously (using a common 5 Vdc supply) activates the DLC and causes the PSEN pin to source 12 Vdc. If this method is chosen, the DLC is not affected by bus activity. **Figure 2-6** shows BUS, $\overline{\text{INT}}$, $\overline{\text{RST}}$, and PSEN timing relationships. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for more information on parameter specifications.

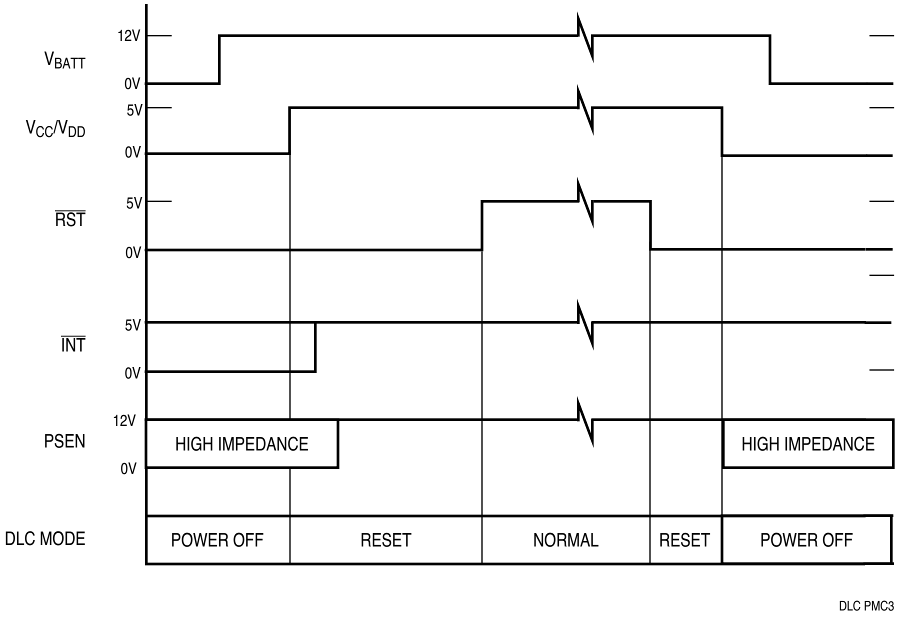


Figure 2-6 Method 3 — Switching V_{DD} and V_{CC} in Standby Mode

2.6 Reset

While the $\overline{\text{RST}}$ signal is asserted, the DLC is held in an inactive state. The DLC is activated when the $\overline{\text{RST}}$ line goes high. When exiting the low-power standby mode, all system power must be applied and stable at the time $\overline{\text{RST}}$ goes high in order for the DLC to be properly activated. During reset, all data in the DLC FIFO buffers is lost.

Many MCUs have low-power standby modes that permit them to remain inactive until needed. These devices typically exit low-power stop when reset, or when an interrupt request is received. If the host MCU, as well as the DLC, are to be activated when the DLC detects J1850 bus activity, use of a common reset line may cause the frame that activates the system to be lost.

After the reset line goes high, the host MCU writes command and configuration bytes to the DLC to initialize it. The “load as configuration byte – immediate” command should be issued to initialize the DLC. The host may commence further operations with the DLC (t_{con}) afterwards. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for more information on parameter specifics. Refer to **SECTION 4 DATA LINK CONTROLLER OPERATION** for more information on system initialization and configuration.

SECTION 3 J1850 FRAME FORMAT

This section explains the frame format used to transmit and receive information on the J1850 bus. Variable pulse width modulation (VPW) valid/invalid bits and symbols are also discussed.

3.1 J1850 Frame Format

A J1850 bus message consists of one or more frames. Each frame is composed of multiple symbols. Refer to **Figure 3-1**.

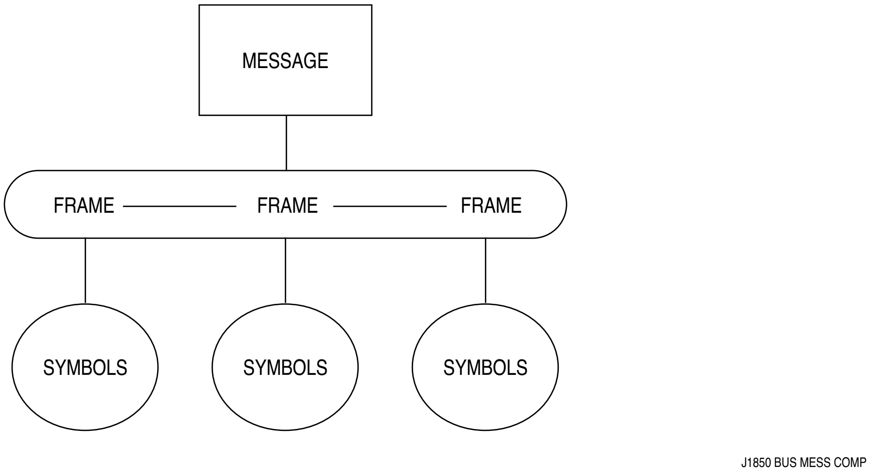


Figure 3-1 J1850 Bus Message Components

A symbol starts with a transition from either an active to a passive level, or a passive to an active level. Symbols end with another transition, or the absence of a transition after a specified time period. **Figure 3-2** illustrates the representative symbols for a J1850 message frame.

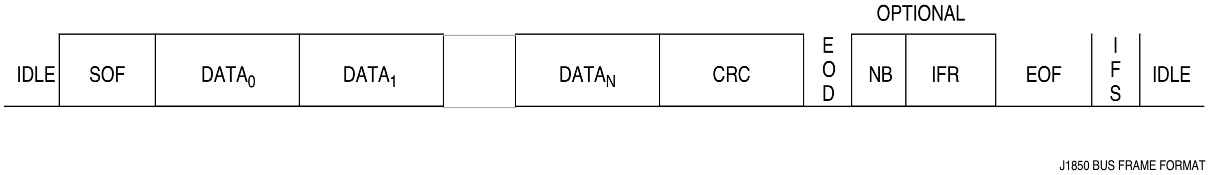


Figure 3-2 J1850 Bus Frame Format (VPW)

Variable pulse width (VPW) modulation is an encoding technique in which each bit is defined by the time between successive transitions, and by the level of the J1850 bus between transitions.

The non-destructive contention protocol on the J1850 bus defines both active and passive symbols. Active and passive bits are used alternately. A symbol is active when one or more transmitters drive the J1850 bus. A symbol is passive when no transmitters are driving the J1850 bus (a logical wired-OR arrangement).

Each logic one or logic zero contains a single transition, and can be at either the active or passive level and one of two lengths, either 64 µs or 128 µs (T_{NOM} at 10.4 kbps baud rate), depending upon the encoding of the previous bit. The SOF, EOD, EOF and IFS symbols are always encoded at an assigned level and length. For an illustration of VPW symbol timing, refer to **Figure 3-3**.

Each frame has a maximum length of 12 bytes, excluding the start of frame (SOF), end of data (EOD), normalization bit (NB), and end of frame (EOF) symbols.

Each frame begins with an SOF symbol, an active symbol, and therefore each data byte (including the CRC byte) begins with a passive bit, regardless of whether it is a logic one or a logic zero.

All VPW bit lengths stated in the following descriptions are typical values at a 10.4 kbps bit rate.

3.1.1 SOF — Start of Frame Symbol

All frames transmitted onto the J1850 bus must begin with an SOF symbol. This indicates to any listeners on the J1850 bus the start of a new frame transmission. The SOF symbol is not used in the cyclical redundancy check (CRC) calculation.

The SOF symbol is defined as a passive to active transition followed by an active period 200 µs in length. Refer to **Figure 3-3** (C). This allows the data bytes which follow the SOF symbol to begin with a passive bit, regardless of whether it is a logic one or a logic zero.

3.1.2 Data — In Frame Data Bytes

The data bytes contained in the frame include the frame header bytes and any actual data being transmitted to the receiving node. The DLC can be used to transmit frames using any of the header formats outlined in the SAE J1850 document. Refer to the **SAE J1850 – CLASS B DATA COMMUNICATIONS NETWORK INTERFACE** for more information about J1850 header formats.

Each data byte is made up of a series of logic one and logic zero symbols. Frames transmitted by the DLC onto the J1850 bus must contain at least one data byte, and therefore can be as short as one data byte and one CRC byte. Each data byte in the frame is eight bits in length, transmitted most significant bit (MSB) to least significant bit (LSB).

3.1.2.1 Logic Zero

A logic zero is defined as either an active to passive transition followed by a passive period 64 µs in length, or a passive to active transition followed by an active period 128 µs in length. Refer to **Figure 3-3** (A).

3.1.2.2 Logic One

A logic one is defined as either an active to passive transition followed by a passive period 128 µs in length, or a passive to active transition followed by an active period 64 µs in length. Refer to **Figure 3-3** (B).

3.1.3 CRC — Cyclical Redundancy Check Byte

The CRC byte is used by the receiver(s) of each frame to determine if any errors have occurred during the transmission of the frame. The DLC calculates the CRC byte and appends it onto any frames transmitted onto the J1850 bus, and also performs CRC detection on any frames it receives from the J1850 bus.

CRC generation uses the divisor polynomial $X^8+X^4+X^3+X^2+1$. The remainder polynomial is initially set to all ones, and then each byte in the frame after the SOF symbol is serially processed through the CRC generation circuitry. The one's complement of the remainder then becomes the 8-bit CRC byte, which is appended to the frame after the data bytes, in MSB to LSB order.

When receiving a frame, the DLC uses the same divisor polynomial. All data bytes, excluding the SOF and EOD symbols, but including the CRC byte, are used to check the CRC. If the frame is error free, the remainder polynomial equals $X^7+X^6+X^2$ ($C4), regardless of the data contained in the frame. If the calculated CRC does not equal $C4, the DLC informs the CPU of the failure.

3.1.4 EOD — End of Data Symbol

The EOD symbol is a passive period on the J1850 bus used to signify to any recipients of a frame that the transmission by the originator has been completed.

The EOD symbol is defined as an active to passive transition followed by a passive period 200 µs in length. Refer to **Figure 3-3** (D and E).

3.1.5 NB — Normalization Bit

The NB is used to preface the in-frame response (IFR). The NB ensures that the end of the eighth bit of the IFR always returns the bus to the passive level. The length of the NB may be used to signify the type of IFR being used. The NB is transmitted by the node responding to the frame, and it defines the start of the optional response segment, if utilized, of any VPW format frame. The DLC indicates that the IFR being transmitted has a CRC appended by using a logic one (active short) bit. The DLC indicates that it does not contain a CRC by using a logic zero (active long) bit.

NOTE

This method of CRC recognition is the reverse of the method preferred by the SAE J1850.

3.1.6 IFR — In-Frame Response Bytes

A number of options are available in the IFR section of the J1850 frame format. The DLC can send an IFR consisting of one or more bytes, which may be followed by a CRC byte.

3.1.7 EOF — End of Frame Symbol

The EOF symbol is a passive period on the J1850 bus, longer than an EOD symbol, which signifies the end of a frame. Since an EOF symbol is longer than an EOD symbol, if no response is transmitted after an EOD symbol, it becomes an EOF, and the frame is assumed to be completed.

The EOF symbol is defined as an active to passive transition followed by a passive period of at least 280 µs in length. Refer to **Figure 3-3** (E). If there is no IFR byte transmitted after an EOD symbol is transmitted, after another 80 µs the EOD becomes an EOF, indicating the completion of the frame.

3.1.8 IFS — Inter-Frame Separation Symbol

The IFS symbol is a passive period on the J1850 bus which allows proper synchronization between nodes during continuous frame transmission. The IFS symbol is transmitted by a node following the completion of the EOF period.

When the last byte of a frame has been transmitted onto the J1850 bus, and the EOF symbol time has expired, all nodes must then wait for the IFS symbol time to expire before transmitting an SOF, marking the beginning of another frame.

However, if the DLC is waiting for the IFS period to expire before beginning a transmission and a passive to active level is detected before the IFS time has expired, it must internally synchronize to that edge. A passive to active level may occur during the IFS period because of varying clock tolerances and loading of the J1850 bus, causing different nodes to observe the completion of the IFS period at different times. Receivers must synchronize to any SOF occurring during an IFS period to allow for individual clock tolerances.

The IFS symbol is defined as an active to passive transition followed by a passive period 300 µs in length. Refer to **Figure 3-3** (E).

3.1.9 BREAK — Break

Any DLC transmitting at the time a BREAK symbol is detected halts transmission immediately, and indicates to the host MCU that a BREAK was detected. The DLC can also transmit a BREAK symbol if necessary.

The BREAK signal is defined as a passive to active transition followed by an active period of at least 239 µs. Refer to **Figure 3-3** (F).

3.1.10 Idle Bus

An idle condition exists on the J1850 bus after expiration of the IFS period. Any node sensing an idle bus condition can begin transmission immediately.

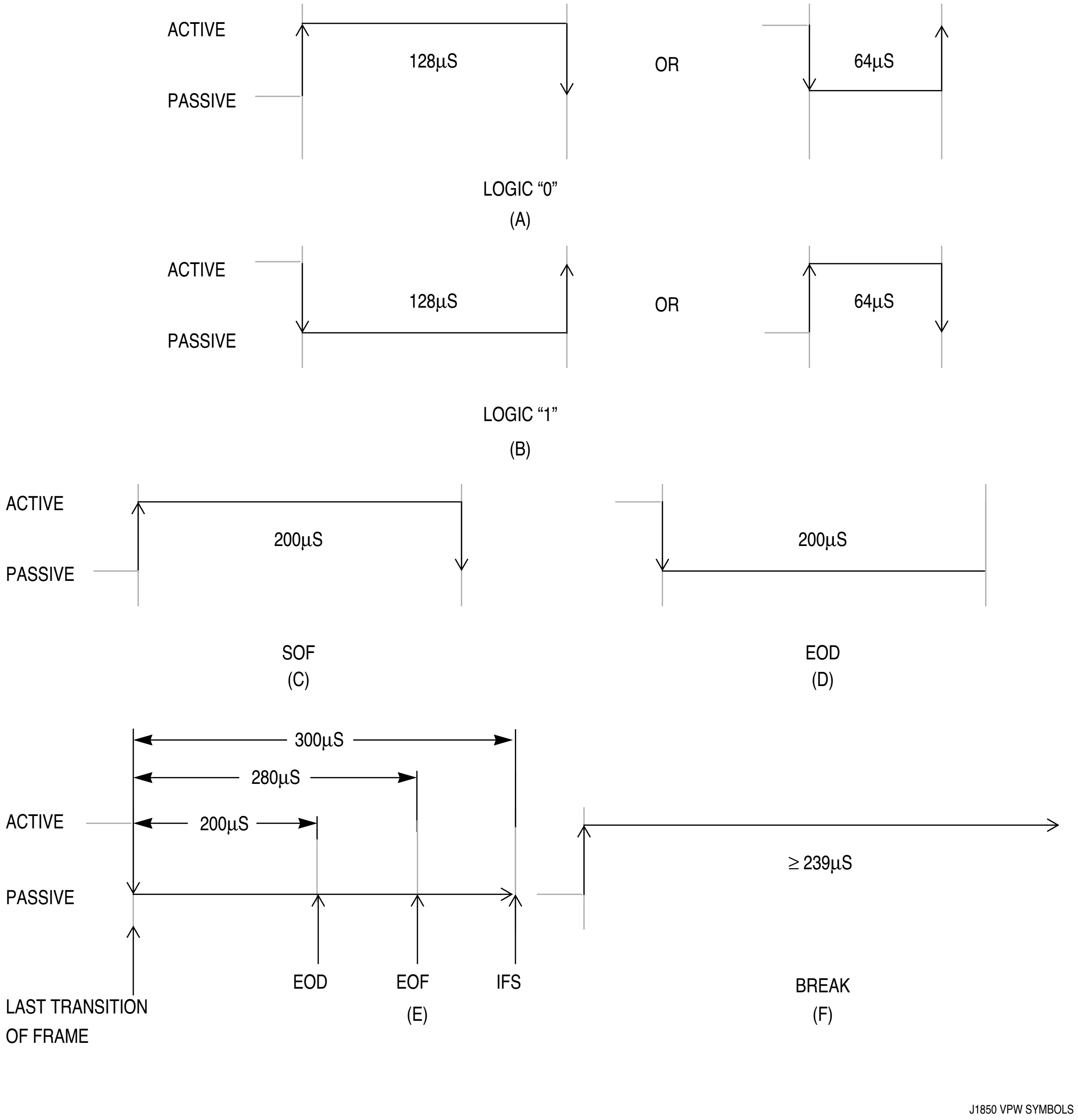


Figure 3-3 J1850 VPW Symbols

3.2 J1850 VPW Valid/Invalid Bits and Symbols

The timing tolerances for receiving data bits and symbols from the J1850 bus allow for variations in the system. The tolerances are balanced by making the maximum of one symbol length approximately equal to the minimum length of the next longest symbol. The difference between the symbol boundaries is equal to one clock period of the DLC internal clock.

Figure 3-4 displays the J1850 passive to active transition symbols. **Figure 3-5** displays the J1850 active to passive transition symbols.

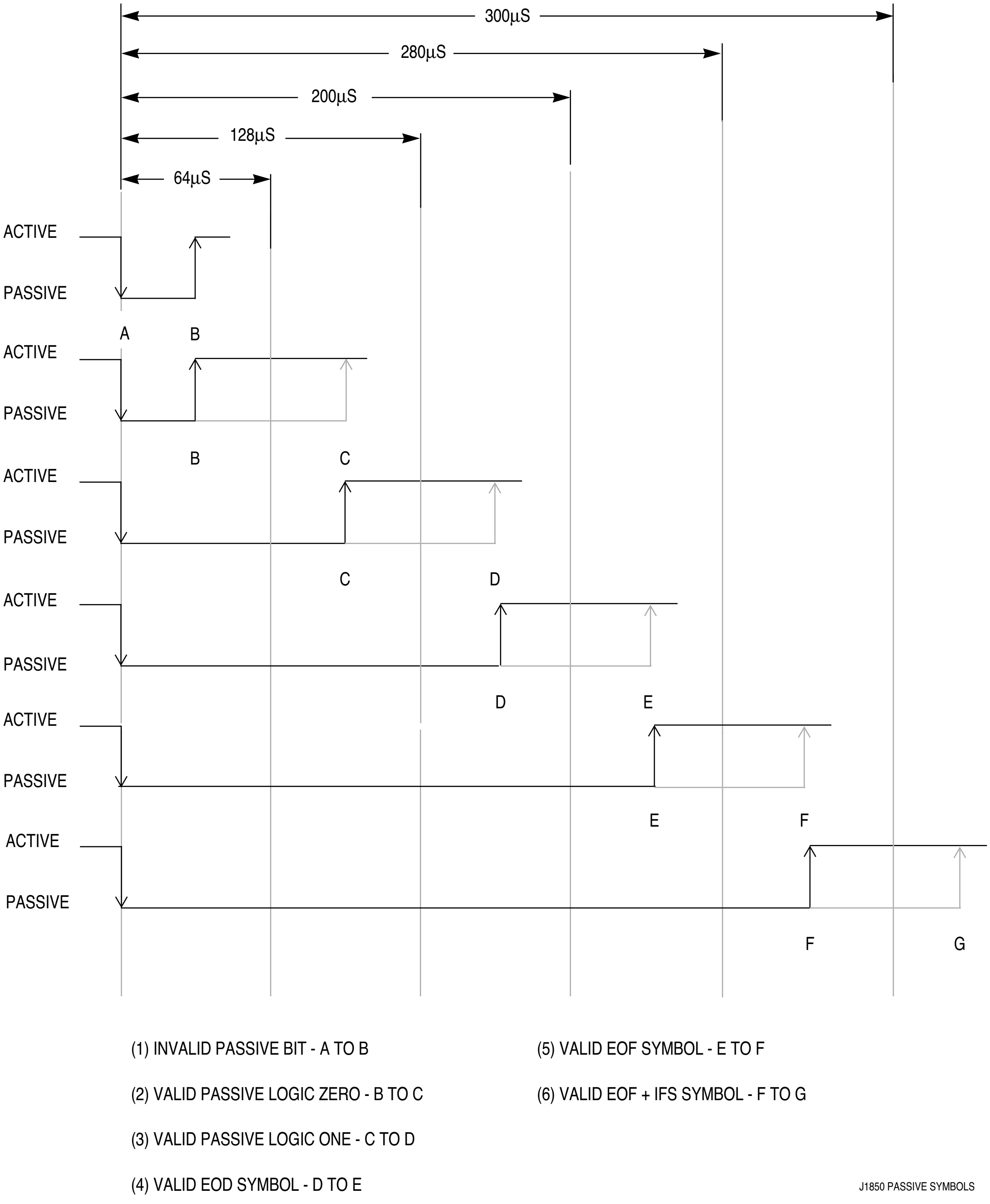

300µS
280µS
200µS
128µS
64µS
ACTIVE
PASSIVE
A
B
ACTIVE
PASSIVE
B
C
ACTIVE
PASSIVE
C
D
ACTIVE
PASSIVE
D
E
ACTIVE
PASSIVE
E
F
ACTIVE
PASSIVE
F
G
(1) INVALID PASSIVE BIT - A TO B
(2) VALID PASSIVE LOGIC ZERO - B TO C
(3) VALID PASSIVE LOGIC ONE - C TO D
(4) VALID EOD SYMBOL - D TO E
(5) VALID EOF SYMBOL - E TO F
(6) VALID EOF + IFS SYMBOL - F TO G
J1850 PASSIVE SYMBOLS

Figure 3-4 J1850 VPW Passive Symbols

3.2.1 Invalid Passive Bit

If the passive to active transition beginning the next data bit or symbol occurs between A (the active to passive transition beginning the current data bit or symbol) and B, the current bit is invalid. Refer to **Figure 3-4** (1).

3.2.2 Valid Passive Logic Zero

If the passive to active transition beginning the next data bit or symbol occurs between B and C, the current bit is considered a logic zero. Refer to **Figure 3-4** (2).

3.2.3 Valid Passive Logic One

If the passive to active transition beginning the next data bit or symbol occurs between C and D, the current bit is considered a logic one. Refer to **Figure 3-4** (3).

3.2.4 Valid EOD Symbol

If the passive to active transition beginning the next data bit or symbol occurs between D and E, the current symbol is considered a valid EOD symbol. Refer to **Figure 3-4** (4).

3.2.5 Valid EOF and IFS Symbol

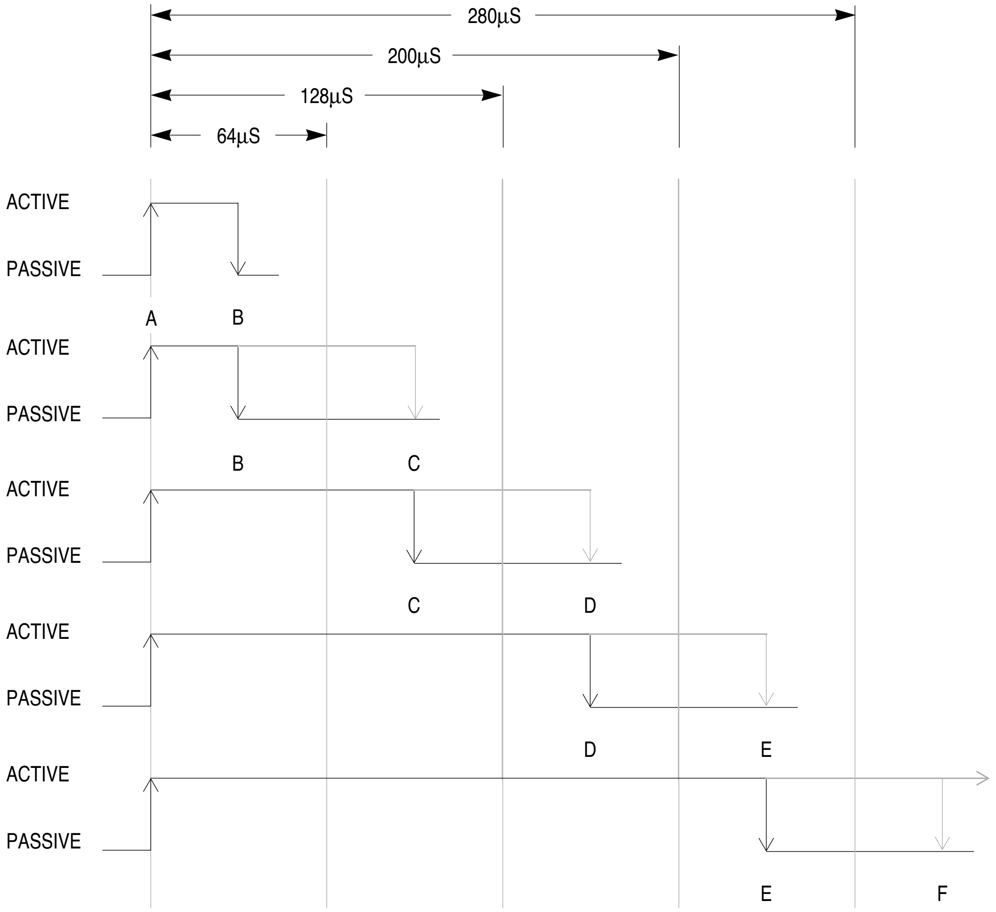
If the passive to active transition beginning the SOF symbol of the next frame occurs between E and F, the current symbol is considered a valid EOF symbol. If the passive to active transition beginning the SOF symbol of the next frame occurs between F and G, the current symbol is considered a valid EOF symbol, followed by a valid IFS symbol. All nodes must wait until a valid IFS symbol time has expired before beginning transmission. However, due to variations in clock frequencies and J1850 bus loading, some nodes may recognize a valid IFS symbol before others, and immediately begin transmitting. Therefore, any time a node waiting to transmit detects a passive to active transition once a valid EOF has been detected, it should immediately begin transmission, initiating the arbitration process. Refer to **Figure 3-4** (5 and 6).

3.2.6 Idle Bus

If the passive to active transition beginning the SOF symbol of the next frame does not occur before G, the J1850 bus is considered to be idle, and any node wishing to transmit a frame may do so immediately. Refer to **Figure 3-4** (6).

3.2.7 Invalid Active Bit

If the active to passive transition beginning the next data bit or symbol occurs between A (the passive to active transition beginning the current data bit or symbol) and B, the current bit is invalid. Refer to **Figure 3-5** (1).



(1) INVALID ACTIVE BIT - A TO B

(2) VALID ACTIVE LOGIC ONE - B TO C

(3) VALID ACTIVE LOGIC ZERO - C TO D

(4) VALID SOF SYMBOL - D TO E

(5) VALID BREAK SYMBOL - E TO F

J1850 ACTIVE SYMBOLS

Figure 3-5 J1850 VPW Active Symbols

3.2.8 Valid Active Logic One

If the active to passive transition beginning the next data bit or symbol occurs between B and C, the current bit is considered a logic one. Refer to **Figure 3-5** (2).

3.2.9 Valid Active Logic Zero

If the active to passive transition beginning the next data bit or symbol occurs between C and D, the current bit is considered a logic zero. Refer to **Figure 3-5** (3).

3.2.10 Valid SOF Symbol

If the active to passive transition beginning the next data bit or symbol occurs between D and E, the current symbol is considered a valid SOF symbol. Refer to **Figure 3-5** (4).

3.2.11 Valid BREAK Symbol

If the next active to passive transition does not occur between E and F, the current symbol is considered a valid BREAK symbol. Following the BREAK symbol, an IFS period must be observed, after which normal communication can resume on the J1850 bus. Refer to **Figure 3-5** (5).

3.3 Frame Arbitration

Frame arbitration on the J1850 bus is accomplished in a non-destructive manner, allowing the frame with the highest priority to be transmitted. Transmitters which lose arbitration simply stop transmitting and wait for an idle J1850 bus to begin transmitting again.

If the DLC wishes to transmit onto the J1850 bus, but detects that another frame is in progress, it must wait until the J1850 bus is idle. However, if multiple nodes begin to transmit in the same synchronization window, frame arbitration occurs beginning with the first bit after the SOF symbol, and continues with each bit thereafter.

The VPW symbols and J1850 bus electrical characteristics are carefully chosen so that a logic zero (active or passive type) always dominates over a logic one (active or passive type) simultaneously transmitted. Hence logic zeros are said to be “dominant” and logic ones are said to be “recessive”. Whenever a node detects a dominant bit when it transmitted a recessive bit, it loses arbitration, and immediately stops transmitting. This is known as “bitwise arbitration”. Refer to **Figure 3-6**.

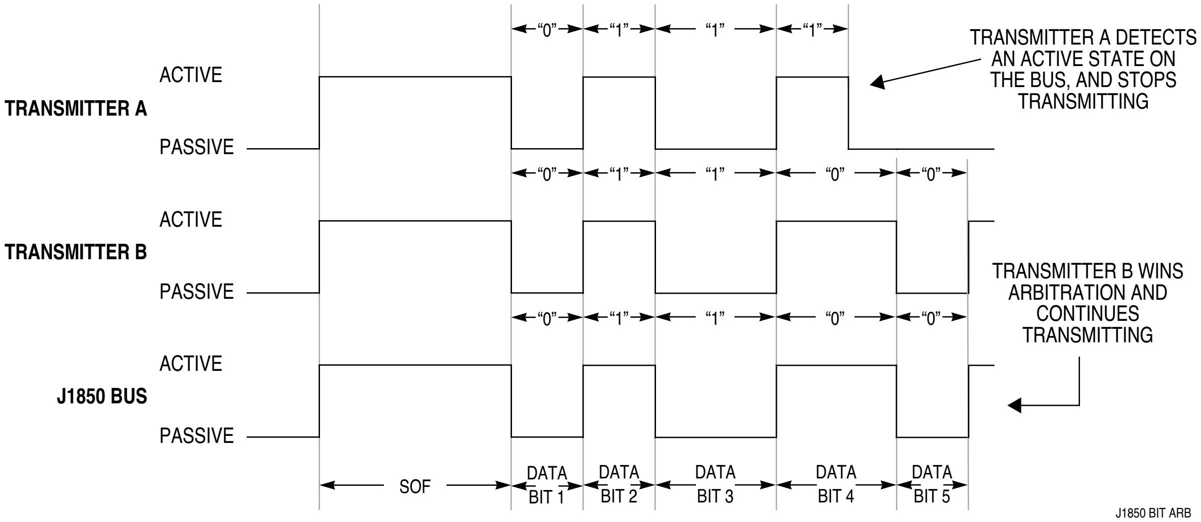


Figure 3-6 J1850 VPW Bitwise Arbitration

During arbitration, or even when the frame is being transmitted, if an opposite bit (a dominant bit when a recessive bit is transmitted, or a recessive bit when a dominant bit is transmitted) is detected, transmission is immediately stopped unless it occurs on the eighth bit of the fourth or later bytes. In this case, the DLC automatically appends up to two extra 1 bits, then stops transmitting. These two extra 1 bits are arbitrated normally, and do not interfere with another frame. The second 1 bit is not sent if the first loses arbitration. If the DLC loses arbitration to another valid frame, the two extra 1 bits do not corrupt the current frame. However, if the DLC has lost arbitration due to noise on the J1850 bus, the two extra 1 bits ensure that the current frame is detected and ignored as a noise-corrupted frame.

Since a zero dominates a one, the frame with the lowest value has the highest priority, and always wins arbitration (a frame with priority 000 wins arbitration over a frame with priority 001). This method of arbitration works no matter how many bits of priority encoding are contained in the frame.

If the DLC loses arbitration during the transmission of a frame, it attempts to retransmit as soon as a valid IFS is detected on the J1850 bus. The DLC attempts to retransmit the frame indefinitely, as long as frames with higher priorities continue to win arbitration. The host MCU can terminate the transmission by transmitting the “terminate auto retry” command to the DLC.

SECTION 4 DATA LINK CONTROLLER OPERATION

This section explains the operation of the MC68HC58 DLC. SAE J1850 bus interface and general operation is the same for both the parallel and serial modes, but the method of transferring information between the host MCU and the DLC differs. Separate discussions of serial and parallel host MCU interfaces follow the general topics. **SECTION 5 CONTROL AND STATUS CODES** examines the specific information exchanged by the host MCU and the DLC.

4.1 Operating Modes

The DLC has six main modes of operation which interact with the power supplies, the J1850 bus interface, and the host MCU interface. Refer to **Figure 4-1**.

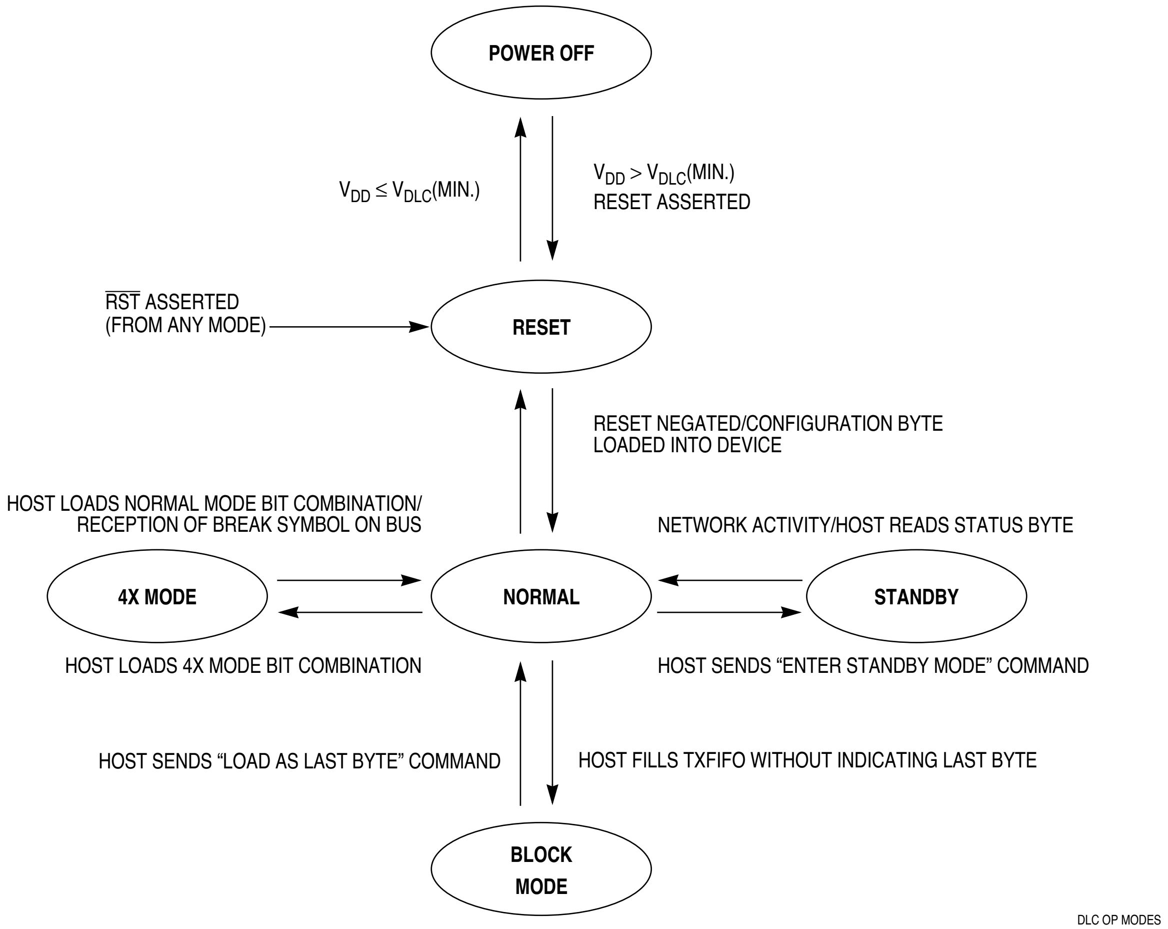

POWER OFF
$V_{DD} \leq V_{DLC}$(MIN.)
$V_{DD} > V_{DLC}$(MIN.)
RESET ASSERTED
$\overline{RST}$ ASSERTED
(FROM ANY MODE)
RESET
RESET NEGATED/CONFIGURATION BYTE
LOADED INTO DEVICE
HOST LOADS NORMAL MODE BIT COMBINATION/
RECEPTION OF BREAK SYMBOL ON BUS
NETWORK ACTIVITY/HOST READS STATUS BYTE
4X MODE
NORMAL
STANDBY
HOST LOADS 4X MODE BIT COMBINATION
HOST SENDS “ENTER STANDBY MODE” COMMAND
HOST SENDS “LOAD AS LAST BYTE” COMMAND
HOST FILLS TXFIFO WITHOUT INDICATING LAST BYTE
BLOCK
MODE
DLC OP MODES

Figure 4-1 DLC Operating Modes

4.1.1 Power-Off Mode

Power-off mode is entered whenever the DLC digital supply voltage V_{DD} drops below the minimum specified value to guarantee correct DLC operation. This includes any standby mode where the V_{DD} supply is switched off. In this mode, the input and output specifications of the host interface signals are not guaranteed.

4.1.2 Reset Mode

Reset mode is entered from the power-off mode whenever the DLC supply voltage V_{DD} rises above the minimum specified value and the DLC $\overline{RST}$ input is asserted.

Reset mode is also entered from any other mode as soon as the DLC $\overline{RST}$ is asserted.

During reset mode, the internal DLC voltage references are operative. V_{DD} is supplied to the internal circuits, which are held in their reset state. The internal DLC system clock continues to run. Registers assume their reset condition. Outputs are held in their programmed reset state, inputs and network activity are ignored.

4.1.3 Normal Mode

Normal mode is entered from the reset mode after the DLC $\overline{RST}$ is negated, and the host loads a configuration byte into the DLC.

Normal mode is entered from the standby mode whenever network activity is sensed, or the host reads the DLC status byte.

Normal mode is entered from the 4X mode when a BREAK symbol is received from the J1850 bus, or the host MCU clears the 4X mode bit in the configuration register.

Normal mode is entered from block mode when the host sends the “load as last byte of transmit data” command.

During normal mode, normal network operation takes place. The user should ensure that all DLC transmissions have ceased before exiting this mode.

4.1.4 Standby Mode

Standby mode is entered when the host sends an “enter standby mode” command.

During standby mode, the DLC internal clocks are halted and the physical interface circuitry is placed in a low power mode to await network or host activity.

4.1.5 4X Mode

4X mode is entered when the host loads the 4X mode bit configuration into the configuration byte.

During 4X mode, transceiver waveshaping is disabled, thus allowing the DLC to operate without any slew rate limitation. 4X mode affects only the transmitted and received symbol timing logic of the DLC (including the digital filter).

4X mode allows communication on the multiplex bus to be performed at four times the normal bit rate.

This high speed transfer function does not work properly unless all transmitting nodes on the J1850 bus are in 4X mode. Selected receivers may be configured to ignore high speed frames until a BREAK symbol is received. Detection of a BREAK symbol on the J1850 bus returns the DLC to normal mode.

Refer to **4.6 BREAK Operation** for more information.

4.1.6 Block Mode

Block mode is entered when the host fills the transmit first in/first out (TxFIFO) buffer without indicating the last byte of a message.

During block mode, the terminate auto retry function is automatically enabled. The DLC continues to transmit message bytes as long as the host loads message bytes into the TxFIFO buffer without sending the DLC a “load as last byte of transmit data” command.

The receive first in/first out (RxFIFO) buffer is designed to allow the reception of block mode messages automatically.

Refer to **4.5 Block Mode Operation** for information.

4.2 Host Interface

Figure 4-2 shows typical usage of a DLC with an MCU.

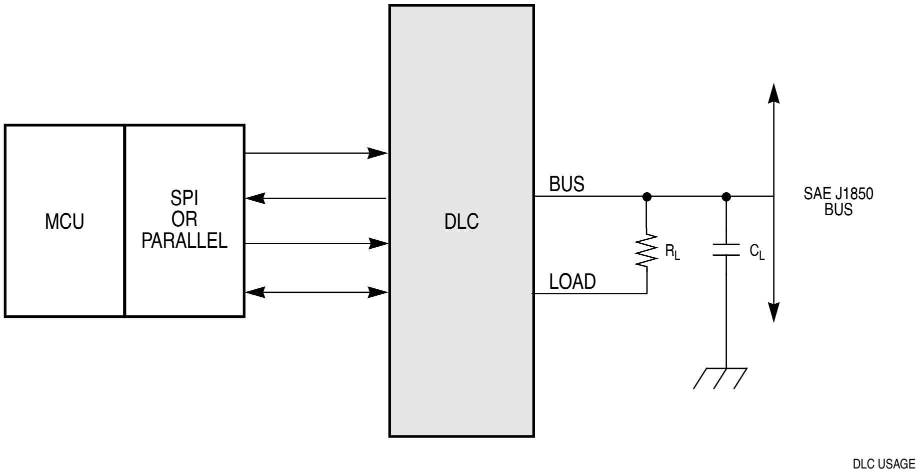


Figure 4-2 DLC Usage

The data link controller exchanges information in pairs of bytes in both the serial and parallel modes. A standard exchange consists of one status byte and one data byte sent from the DLC to the host, and one data byte and one command byte sent from the host to the DLC. In serial mode, the host and the DLC exchange pairs of bytes simultaneously, while exchanges in parallel mode are sequential. A host can write contiguous data bytes to the DLC in parallel mode without intervening command bytes, provided the first and last data bytes are accompanied by “load as first byte of transmit data”, “load as last byte of transmit data”, or “load as first and last byte of transmit data” command bytes.

Data that the DLC receives from the host can be data for transmission on the SAE J1850 bus, a configuration byte, or a null byte. The destination of a byte written to a DLC is specified in the command byte that accompanies it. Command bytes can also contain DLC receive, transmit, and mode control instructions.

The status byte that the host receives from the DLC contains information about the accompanying data byte, current status of the transmitter, and current status of the receiver. The data byte can contain received frame data, a transmission completion code, or invalid information.

The DLC must be configured after power-up or reset, and at appropriate intervals thereafter. To configure a DLC, the host MCU sends a configuration command byte followed by a configuration data byte. Mode of operation, clock speed, and interrupt mode are determined by the content of the configuration byte.

When the $\overline{\text{CS}}$ signal is asserted, interface lines (parallel or serial) are enabled, then status and data words are prepared for transfer to the host. Parallel transfers are controlled by the ADDR0, CLK, and R/$\overline{\text{W}}$ signals. Serial transfers are clocked by the SCLK signal. When $\overline{\text{CS}}$ is released, interface lines are disabled and go to a high-impedance state.

When interrupts are enabled, the DLC generates an interrupt based on four, or optionally five interrupt sources. When one of these interrupt conditions occurs, the DLC asserts the $\overline{\text{INT}}$ output to request service from the host. The $\overline{\text{INT}}$ output is negated when $\overline{\text{CS}}$ is asserted by the host and the status byte is read. Refer to **4.2.3 Interrupt Requests** for more information.

Figure 4-3 shows typical DLC operation.

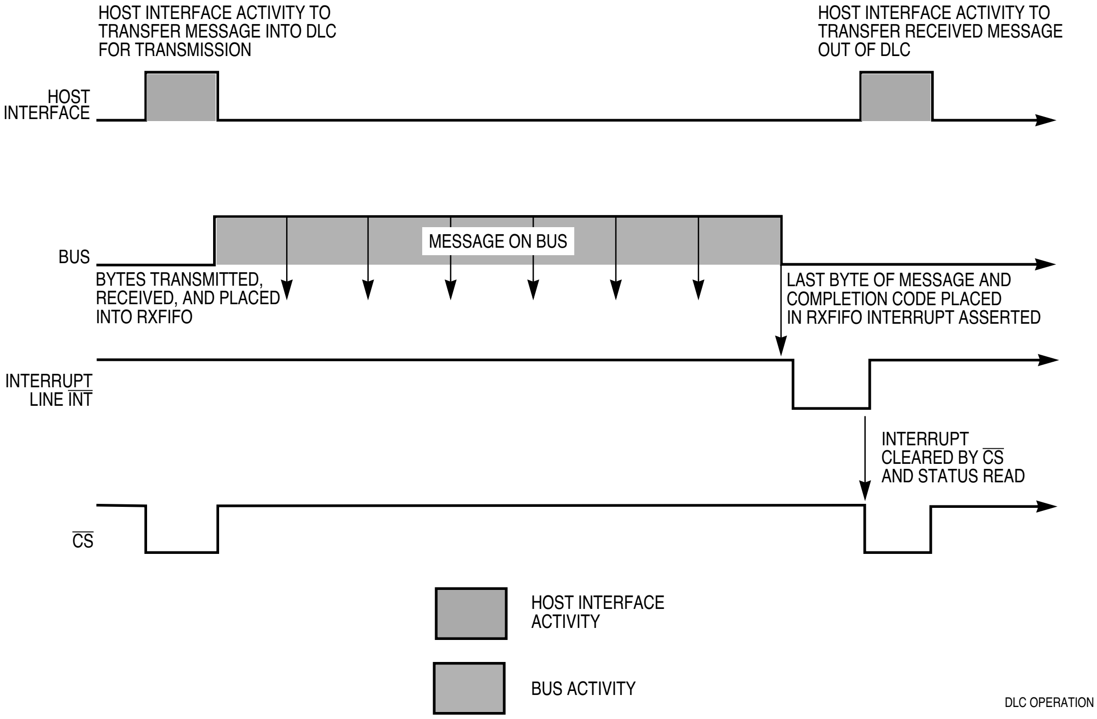

HOST INTERFACE ACTIVITY TO TRANSFER MESSAGE INTO DLC FOR TRANSMISSION
HOST INTERFACE ACTIVITY TO TRANSFER RECEIVED MESSAGE OUT OF DLC
HOST INTERFACE
MESSAGE ON BUS
BUS
BYTES TRANSMITTED, RECEIVED, AND PLACED INTO RXFIFO
LAST BYTE OF MESSAGE AND COMPLETION CODE PLACED IN RXFIFO INTERRUPT ASSERTED
INTERRUPT LINE $\overline{INT}$
INTERRUPT CLEARED BY $\overline{CS}$ AND STATUS READ
$\overline{CS}$
HOST INTERFACE ACTIVITY
BUS ACTIVITY
DLC OPERATION

Figure 4-3 DLC Operation

The host MCU can put a DLC in standby mode by sending it the “enter standby mode” command byte. The DLC completes all current frames, then goes inactive. PSEN sources V_{BATT} while the DLC is active and goes to a high-impedance state during standby mode. An incoming bus frame, or the host MCU asserting $\overline{CS}$ and reading the status byte, can reactivate a DLC.

4.2.1 MC68HC58 DLC Parallel Mode Host MCU Interface

The DLC interface is designed to work with an M6800 bus clock signal (CLK), but functions equally well with any host MCU that meets timing requirements. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for more parallel host interface timing information.

4.2.1.1 Parallel Mode Data Transfer

Parallel mode transfers are controlled by the R/$\overline{W}$ and ADDR0 signals from the host MCU. R/$\overline{W}$ and ADDR0 determine which of four possible bytes are being transferred. **Figure 4-4** shows the parallel mode byte format.

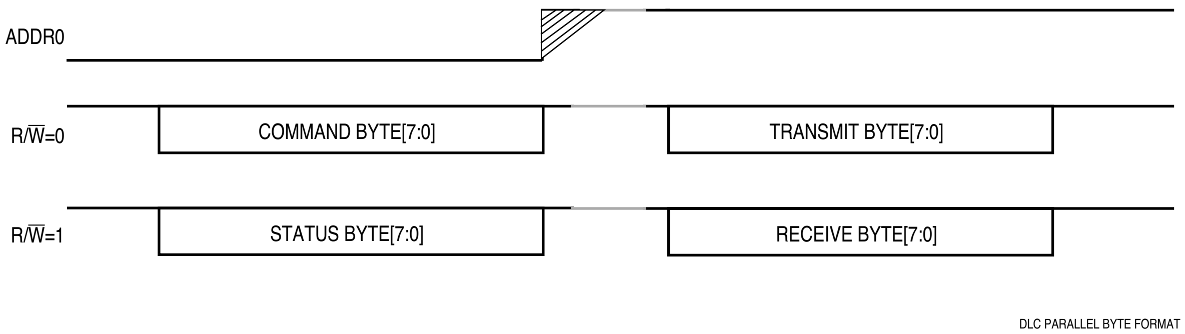


Figure 4-4 Parallel Mode Byte Format

DLC command bytes are transferred before the accompanying data byte. When using the parallel host interface, the DLC requires command bytes only for the first and last byte(s) in a frame. A DLC command byte must be accompanied by another byte (data, null, or configuration) in order for the command to be latched. A command byte is not required to load each data byte into the TxFIFO buffer, but a data byte is required for a command to execute. **Table 4-1** shows the types of parallel transfers.

Table 4-1 Parallel Transfers

ADDR0	R/$\overline{W}$	Type of Transfer
0	0	Command byte
0	1	Status byte
1	0	Transmit byte or configuration byte
1	1	Receive byte or completion code

The status byte value is frozen while $\overline{CS}$ is asserted. Any changes that occur while $\overline{CS}$ is asserted are reflected in the status byte when $\overline{CS}$ is next asserted.

The DLC interrupt request line is cleared when $\overline{CS}$ is asserted and the status byte is read. When $\overline{CS}$ is negated, the DLC parallel lines are disabled, and go into a high impedance state.

4.2.1.2 Servicing Sequence

The host MCU services the DLC in the following sequence:

1. Read status byte.
2. Read receive data (if required).
3. Write command byte.
4. Write transmit data, configuration, or null byte (if preceded by command byte).

The four operations form a complete service sequence. However, all four operations are not required in every host/DLC transfer.

The host MCU can read only the status byte, or can read the status byte followed by a read of a data byte (the data read automatically flushes the associated data byte from the RxFIFO buffer). There is no specified maximum time between reading of a status byte and reading of a data byte, but correct status of the RxFIFO is guaranteed only for 1 µs.

The DLC allows a host to write contiguous frame bytes to the TxFIFO buffer without intervening command bytes, provided that the first and last data bytes are accompanied by “load as first byte of transmit data”, “load as last byte of transmit data”, or “load as first and last byte of transmit data” command bytes. A data, configuration, or null byte need not immediately follow the associated command, but must be the next byte transferred to the DLC. The command byte is acted upon when the subsequent byte is written into the DLC.

4.2.1.3 Minimum Time Requirements

Minimum time constraints appear on certain sequences of operations. **Table 4-2** shows the minimum time required between these operations. The left column shows initial operations; the top row, subsequent operations.

Table 4-2 Minimum Time Between Operations

	Write Command	**Write Data/Config**	**Read Status**	**Read Data**
Write Command	Not allowed	0	Not allowed	Not allowed
Write Data/Config	3 µs	3 µs[1]	3 µs	Not allowed
Read Status	0	Not allowed	0[2]	0
Read Data	3 µs	Not allowed	0	Not allowed

NOTES:

1. During block transfers only.
2. Status valid for 1 µs

4.2.1.4 Motorola Microcontroller Data Transfers

Many Motorola microcontroller families, including the M68HC11, M68HC16, and M68300 series MCUs, can generate synchronous ADDR0, DATA, CLK and R/$\overline{W}$ signals.

For example, host MCUs based on M68HC11 microcontrollers can use instructions that load 16-bit double accumulator D from two consecutive byte addresses (LDD), and store accumulator content into two consecutive byte addresses (STD).

Host MCUs based on M68HC16 and M68300 microcontrollers can use the dynamic bus sizing capability to automate the transfer of 16-bit data to and from an 8-bit port.

4.2.2 MC68HC58 DLC Serial Mode Host MCU Interface

In serial mode, the MC68HC58 DLC communicates with the host MCU via the SPI. The SPI performs simultaneous bidirectional transfers initiated by an SPI bus master, with the DLC functioning as an SPI slave device. SPI transfers are timed by the SCLK signal from the host MCU. All Motorola MCU families (M68HC05, M68HC08, M68HC11, M68HC12, M68HC16, M68300 and MPC500) have devices with SPI capabilities, but any host MCU that meets the clock specification can be used. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for more information.

4.2.2.1 Serial Mode Data Transfer

Data is always transferred between the host MCU and the DLC in a two byte format, with an optional delay between bytes for data management. This delay can be omitted if the host is utilizing a 16-bit shift register. The $\overline{\text{CS}}$ signal must remain asserted throughout the duration of the 16-bit transfer.

While the host MCU is transmitting a data byte to the DLC, the DLC is transmitting a status byte to the host MCU. This status byte updates the host MCU as to the status of the DLC and its FIFO buffers. The host MCU then transmits a command byte to the DLC, which tells the DLC what to do with the preceding data byte, while the DLC transmits a data byte to the host MCU. In the DLC, when the RFS field indicates that a completion code is at the head of the buffer, the accompanying data byte contains that same completion code.

The command, status and data bytes are ordered MSB-to-LSB, and are transmitted in the non-return to zero (NRZ) bit format. **Figure 4-5** shows the DLC serial mode byte format.

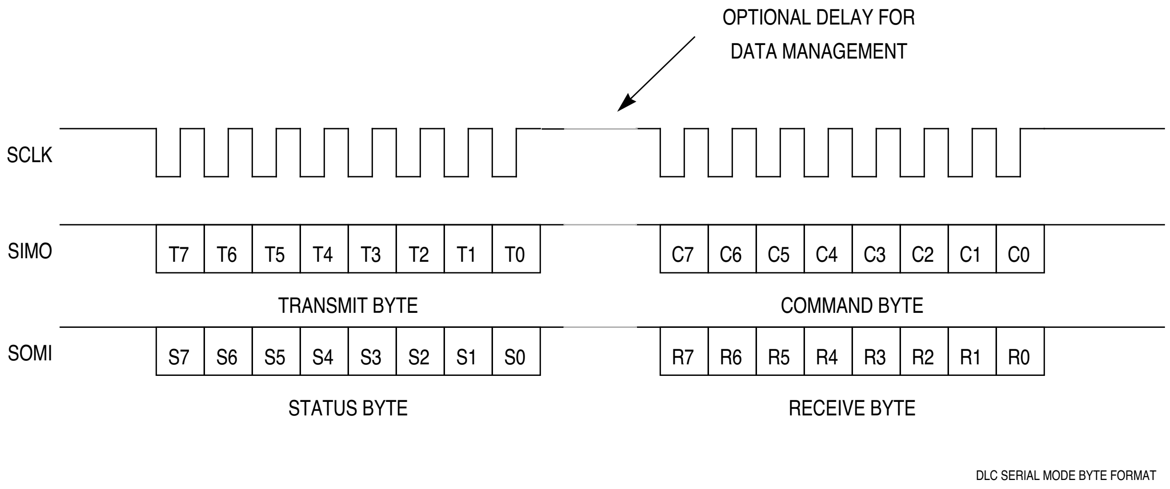


Figure 4-5 DLC Serial Mode Byte Format

Table 4-3 shows the types of serial transfers.

Table 4-3 Serial Transfers

1st Byte	2nd Byte	Type of Transfer
—	SIMO	Command byte
SOMI	—	Status byte
SIMO	—	Transmit byte, configuration byte, or null byte
—	SOMI	Receive byte, completion code, or null byte

4.2.2.2 Servicing Sequence

For every byte transmitted from the host to the DLC, there is a byte transferred from the DLC to the host. Each data byte sent to the DLC is accompanied by a status byte that is sent to the host. Each command byte sent to the DLC is accompanied by a received data byte that is sent to the host.

The host MCU services the DLC in the following sequence:

1. Read status byte and data byte.
2. Write data byte, configuration byte, or null byte and command byte.

Received data remains in the RxFIFO buffer until a flush byte command is received. However, the flush command can accompany the read command. Each transmit data byte must be accompanied by a command byte that designates it as a first, intermediate, or last byte.

4.2.2.3 SPI Exchange

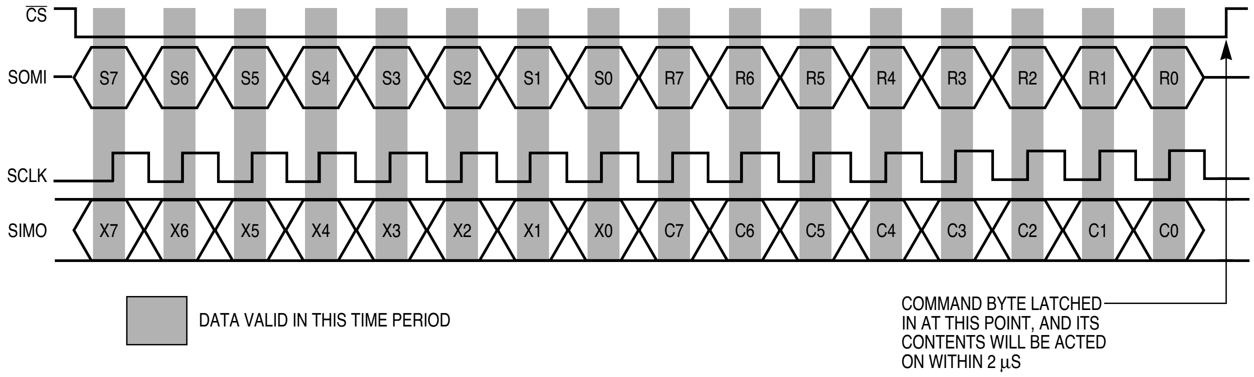
The host MCU controls SCLK polarity and phase. The DLC is compatible with two SCLK configurations:

- SCLK polarity normally high, read data on SCLK passive to active level, change data on SCLK active to passive level.
- SCLK normally low, read data on SCLK passive to active level, change data on SCLK active to passive level.

The general format of the data exchange from the DLC to the host MCU is a bit-for-bit exchange on each SCLK clock pulse. Data is read in on the passive to active level of the SCLK, and is changed on the active to passive level of SCLK. The most significant bit of a transfer is sent first. **Figure 4-6** shows a 16-bit data transfer with low clock polarity. **Figure 4-7** shows a 16-bit data transfer with high clock polarity.

NOTE

This 16-bit transfer can be performed using two 8-bit transfers as long as $\overline{\text{CS}}$ is asserted before the first 8-bit transfer begins, and remains asserted until the end of the second 8-bit transfer.



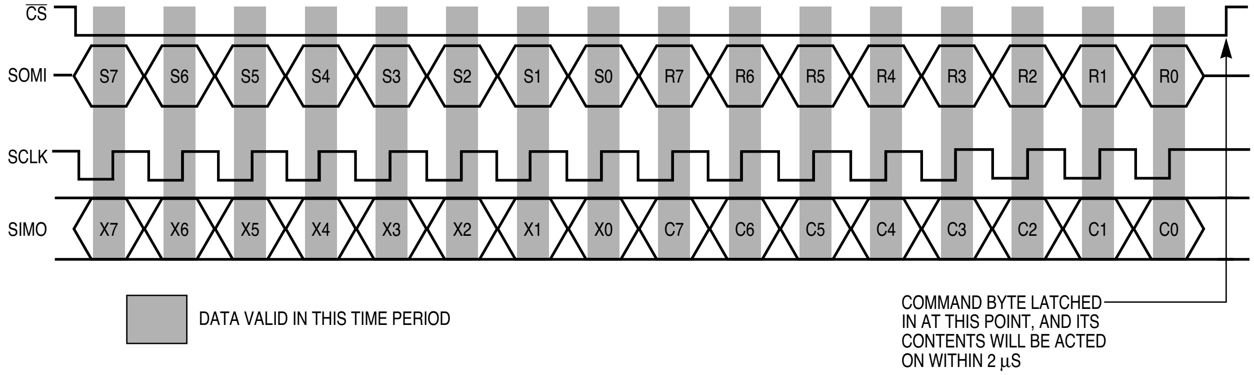
▨ DATA VALID IN THIS TIME PERIOD

COMMAND BYTE LATCHED IN AT THIS POINT, AND ITS CONTENTS WILL BE ACTED ON WITHIN 2 µS

NOTE: SCLK NORMALLY LOW (POLARITY = 0), PHASE = 0

DLC SPI TRANSFER 1

Figure 4-6 SPI Transfer — Clock Polarity Low



▨ DATA VALID IN THIS TIME PERIOD

COMMAND BYTE LATCHED IN AT THIS POINT, AND ITS CONTENTS WILL BE ACTED ON WITHIN 2 µS

NOTE: SCLK NORMALLY HIGH (POLARITY = 1), PHASE = 1

DLC SPI TRANSFER 2

Figure 4-7 SPI Transfer — Clock Polarity High

4.2.2.4 Initialization

Before the DLC is used for transmission or reception, it must be initialized. Initialization is performed by loading a configuration data byte into the configuration byte register with the “load as configuration byte—immediate” command. Once the configuration byte is loaded following a reset of the DLC, the host can load message bytes to be transmitted onto the J1850 bus into the DLC. Transmission will not begin until the DLC determines that an idle bus condition exists on the J1850 bus. **Figure 4-8** illustrates a basic host/DLC initialization routine.

NOTE

Use the “load as configuration byte—immediate” command byte bit combination only after a reset, for test purposes, or for changing the interrupt mode or mask during normal operation. In the absence of these conditions, use the “load as configuration byte” command.

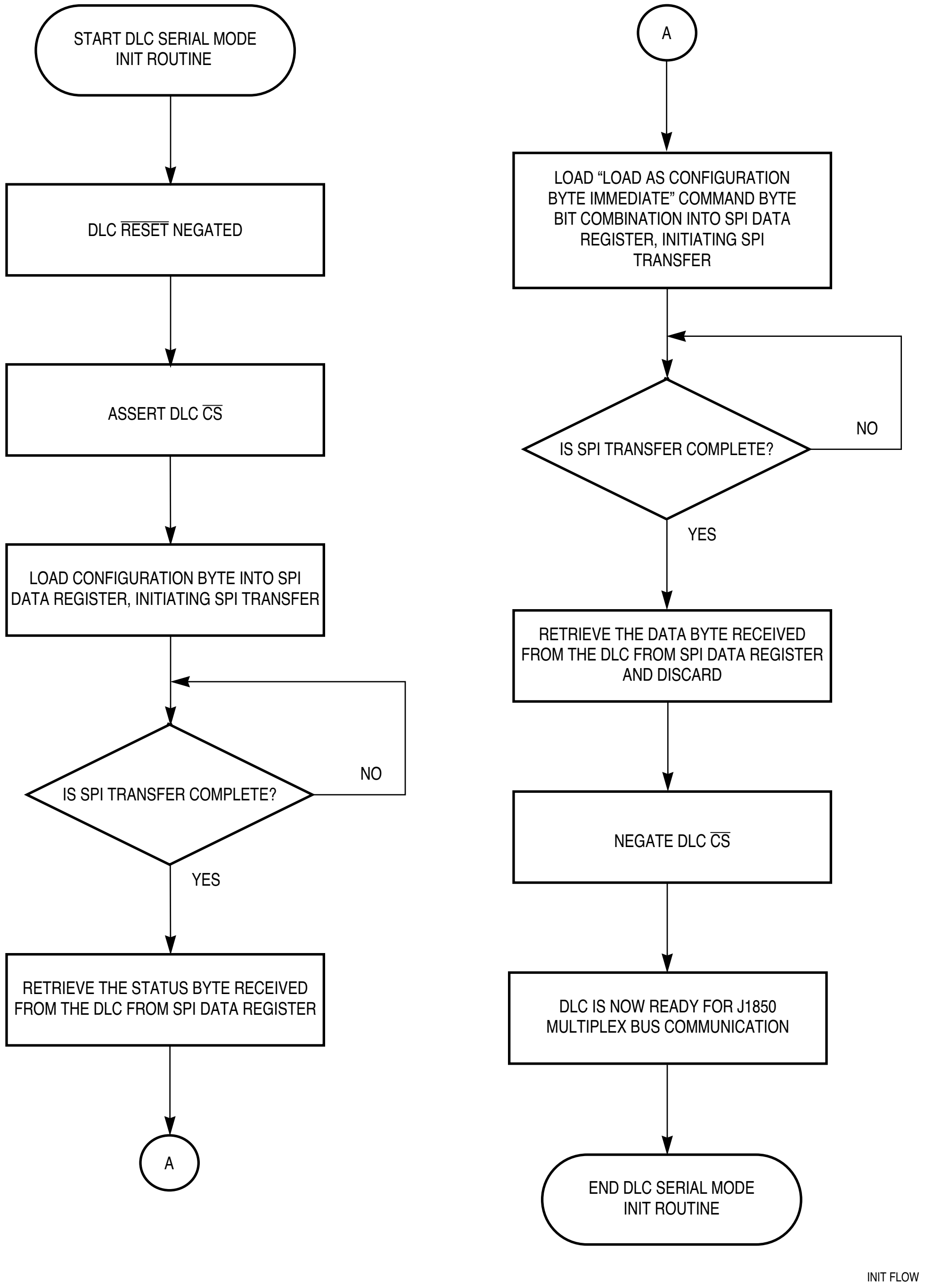

START DLC SERIAL MODE INIT ROUTINE
DLC RESET NEGATED
ASSERT DLC CS
LOAD CONFIGURATION BYTE INTO SPI DATA REGISTER, INITIATING SPI TRANSFER
IS SPI TRANSFER COMPLETE?
NO
YES
RETRIEVE THE STATUS BYTE RECEIVED FROM THE DLC FROM SPI DATA REGISTER
A
A
LOAD “LOAD AS CONFIGURATION BYTE IMMEDIATE” COMMAND BYTE BIT COMBINATION INTO SPI DATA REGISTER, INITIATING SPI TRANSFER
IS SPI TRANSFER COMPLETE?
NO
YES
RETRIEVE THE DATA BYTE RECEIVED FROM THE DLC FROM SPI DATA REGISTER AND DISCARD
NEGATE DLC CS
DLC IS NOW READY FOR J1850 MULTIPLEX BUS COMMUNICATION
END DLC SERIAL MODE INIT ROUTINE
INIT FLOW

Figure 4-8 Host/DLC Serial Mode Initialization Routine

4.2.3 Interrupt Requests

Generation of interrupt requests is determined by the values of the configuration byte IMSK and IMOD bits. By default, the DLC automatically asserts the $\overline{\text{INT}}$ signal when one or more of four conditions occur. Setting the IMSK bit disables interrupts. Setting the IMOD bit adds a fifth interrupt request condition.

When a condition that would normally cause an interrupt request occurs while the chip select input is asserted, the DLC cannot assert $\overline{\text{INT}}$ until the logic level on the $\overline{\text{CS}}$ pin returns to one.

In default configuration, the DLC asserts $\overline{\text{INT}}$ under the following circumstances (a 4 MHz reference is assumed):

- When a frame completion code is placed in the RxFIFO. The DLC asserts $\overline{\text{INT}}$ within 10 µs of sensing EOD on the J1850 bus.
- When the RxFIFO buffer has 12 bytes in it and a 13th byte is received. The DLC asserts $\overline{\text{INT}}$ within 15 µs after the trailing edge of the first bit of the 14th byte received.
- When the TxFIFO is half emptied during a block mode transmission (no byte in the TxFIFO buffer has been accompanied by a “load a last byte of transmit data” command). Six bytes remain to be transmitted when the interrupt request is made. The DLC asserts $\overline{\text{INT}}$ within ten µs after the trailing edge of the last bit of the fifth byte transmitted.
- When the DLC comes out of standby mode due to activity on the SAE J1850 bus. The DLC asserts $\overline{\text{INT}}$ within 105 µs after the passive to active level of the SOF symbol.
- After a configuration byte with IMOD set is sent to a DLC, the DLC also asserts $\overline{\text{INT}}$ when a byte is received into an empty RxFIFO. $\overline{\text{INT}}$ assertion occurs 15 µs or less after the trailing edge of the first bit of the second byte received.

The host MCU must use information in the status byte or the completion code to determine the interrupt source. It cannot deduce the cause of a TxFIFO half empty interrupt during block mode transmission because there is no explicit status indication for a TxFIFO buffer underrun. The host MCU can, however, monitor DLC status bytes for an indication that the TxFIFO buffer is no longer full.

4.3 Transmitter Operation

The DLC transmitter drives the J1850 bus to at least 6.25 Vdc, and expects the external load to pull the J1850 bus down below 1.5 Vdc. The transmitter responds to feedback from the receiver in order to determine precisely when to start and finish driving the bus. A set of basic transmit timing windows for J1850 bus symbols is stored in the DLC logic. The transmitter uses these timing windows as a basis for transmission, but if the receiver detects a change in the J1850 bus state during transmission, the transmitter switches to that state unless the switch causes the transmitter to lose arbitration. If arbitration is lost, the transmitter halts. This mechanism prevents J1850 bus conflict due to slight timing differences between J1850 bus nodes. Only one frame appears on the J1850 bus during a transmission. Symbols in the frame are driven and released by all participating nodes, but the end result is a single waveform.

To transmit a frame on the J1850 bus, a host MCU first transfers frame data to the DLC. For parallel mode, a command byte need only accompany and identify the first and last byte(s). For serial mode, each byte of data must be accompanied by a command byte that tells the DLC whether the data is a first, intermediate, or last byte.

Frames are sent in two basic modes:

- A complete frame is loaded into the TxFIFO buffer for transmission (Normal mode).
- Bytes of the frame are continuously loaded into the TxFIFO buffer as the message is being transmitted, until the entire frame has been sent (Block mode).

The DLC determines that a frame is completely loaded into the TxFIFO by reading a “load as last byte of transmit data” command byte. If no such command byte has been received, the transmitter operates in block mode until the command is received. The host MCU can monitor transmitter activity by polling individual status bytes, or servicing can be interrupt driven. Refer to **4.5 Block Mode Operation** for more information.

The transmitter waits until either an entire frame is in the TxFIFO buffer, or until the TxFIFO buffer is full before beginning a transmission. This prevents a data underrun during transmission of the first frame byte. Once the J1850 bus is determined to be idle, transmission and J1850 bus arbitration begin.

When a complete frame is in the TxFIFO buffer, the DLC status byte will indicate that the TxFIFO buffer is full until the transmission is successfully completed.

A DLC automatically attempts to retransmit a frame if it loses arbitration or if errors are detected during transmission. The completion code placed in the RxFIFO indicates to the host MCU when a frame has lost arbitration.

The host MCU can terminate this automatic retry by writing a “terminate automatic retry ” (TAR) command byte to the DLC. When the DLC receives the TAR command, it completes any current transmission, then clears the TxFIFO buffer. If no transmission is in progress, and the TxFIFO buffer is full, the DLC attempts to transmit the frame in the TxFIFO buffer once, then clears the TxFIFO buffer.

Since a frame remains in the TxFIFO buffer until it is successfully transmitted, a host must clear the TxFIFO buffer if another frame is to be substituted for it. There are several commands that can accomplish this. Refer to **SECTION 5 CONTROL AND STATUS CODES** for more information. One of these commands and the command to load the first byte of a new frame can be combined into a single byte that accompanies the first byte of new data.

When a TxFIFO underrun occurs, the CRC is intentionally corrupted by being complemented and appended to the transmission.

Figures 4-9 and **4-10** outline the basic software requirements for transferring data to the DLC in serial mode for transmission onto the J1850 multiplex bus.

NOTE

> Particular applications may require more extensive error monitoring and handling routines than this flowchart displays. Conditions reflected in the retrieved status byte (received data bytes or J1850 bus status) are also not addressed.

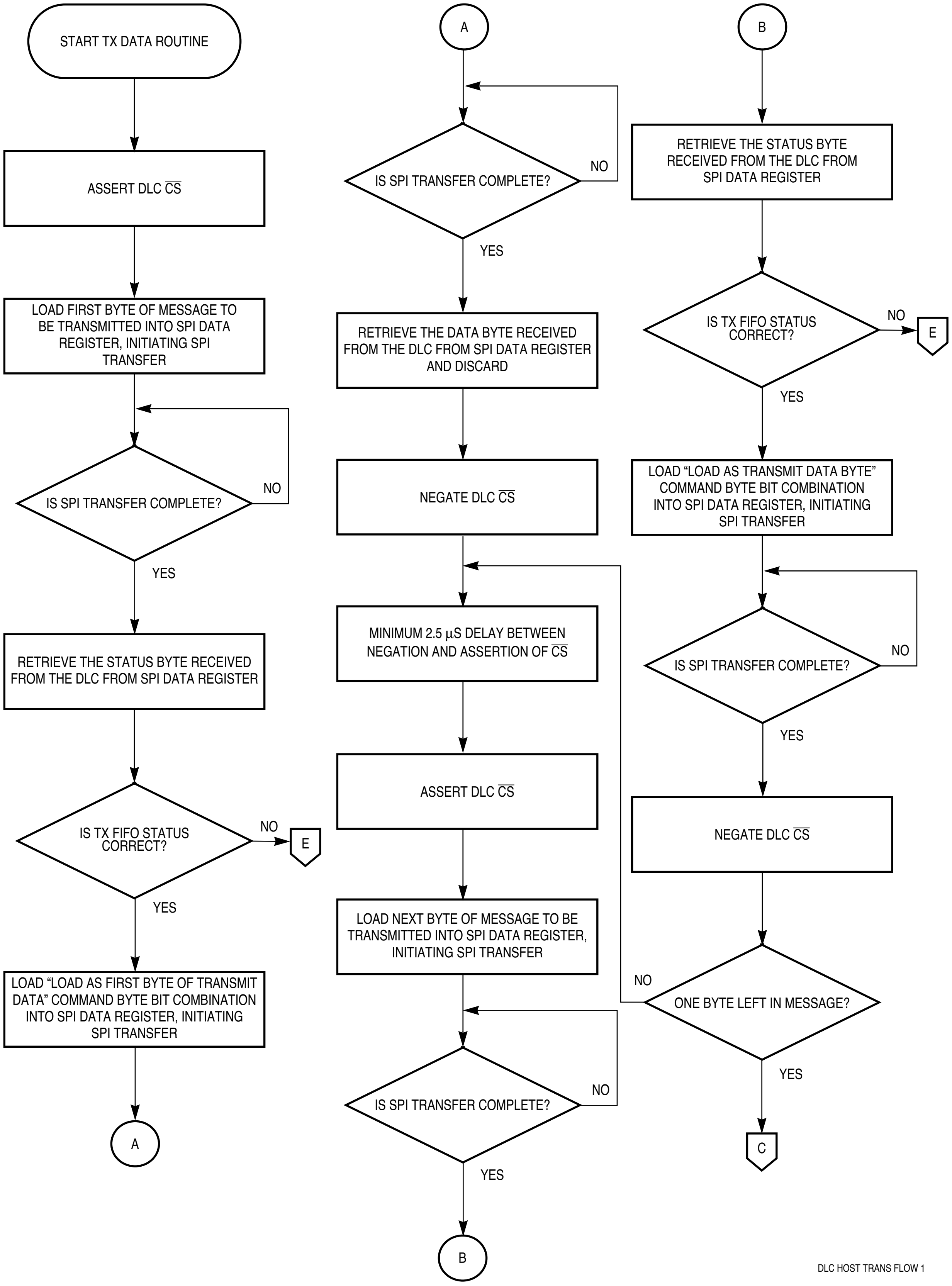

START TX DATA ROUTINE
ASSERT DLC $\overline{CS}$
LOAD FIRST BYTE OF MESSAGE TO BE TRANSMITTED INTO SPI DATA REGISTER, INITIATING SPI TRANSFER
IS SPI TRANSFER COMPLETE?
NO
YES
RETRIEVE THE STATUS BYTE RECEIVED FROM THE DLC FROM SPI DATA REGISTER
IS TX FIFO STATUS CORRECT?
NO
E
YES
LOAD "LOAD AS FIRST BYTE OF TRANSMIT DATA" COMMAND BYTE BIT COMBINATION INTO SPI DATA REGISTER, INITIATING SPI TRANSFER
A
A
IS SPI TRANSFER COMPLETE?
NO
YES
RETRIEVE THE DATA BYTE RECEIVED FROM THE DLC FROM SPI DATA REGISTER AND DISCARD
NEGATE DLC $\overline{CS}$
MINIMUM 2.5 µS DELAY BETWEEN NEGATION AND ASSERTION OF $\overline{CS}$
ASSERT DLC $\overline{CS}$
LOAD NEXT BYTE OF MESSAGE TO BE TRANSMITTED INTO SPI DATA REGISTER, INITIATING SPI TRANSFER
IS SPI TRANSFER COMPLETE?
NO
YES
B
B
RETRIEVE THE STATUS BYTE RECEIVED FROM THE DLC FROM SPI DATA REGISTER
IS TX FIFO STATUS CORRECT?
NO
E
YES
LOAD "LOAD AS TRANSMIT DATA BYTE" COMMAND BYTE BIT COMBINATION INTO SPI DATA REGISTER, INITIATING SPI TRANSFER
IS SPI TRANSFER COMPLETE?
NO
YES
NEGATE DLC $\overline{CS}$
ONE BYTE LEFT IN MESSAGE?
NO
YES
C
DLC HOST TRANS FLOW 1

Figure 4-9 Host/DLC Serial Mode Transmit Routine (Part 1 of 2)

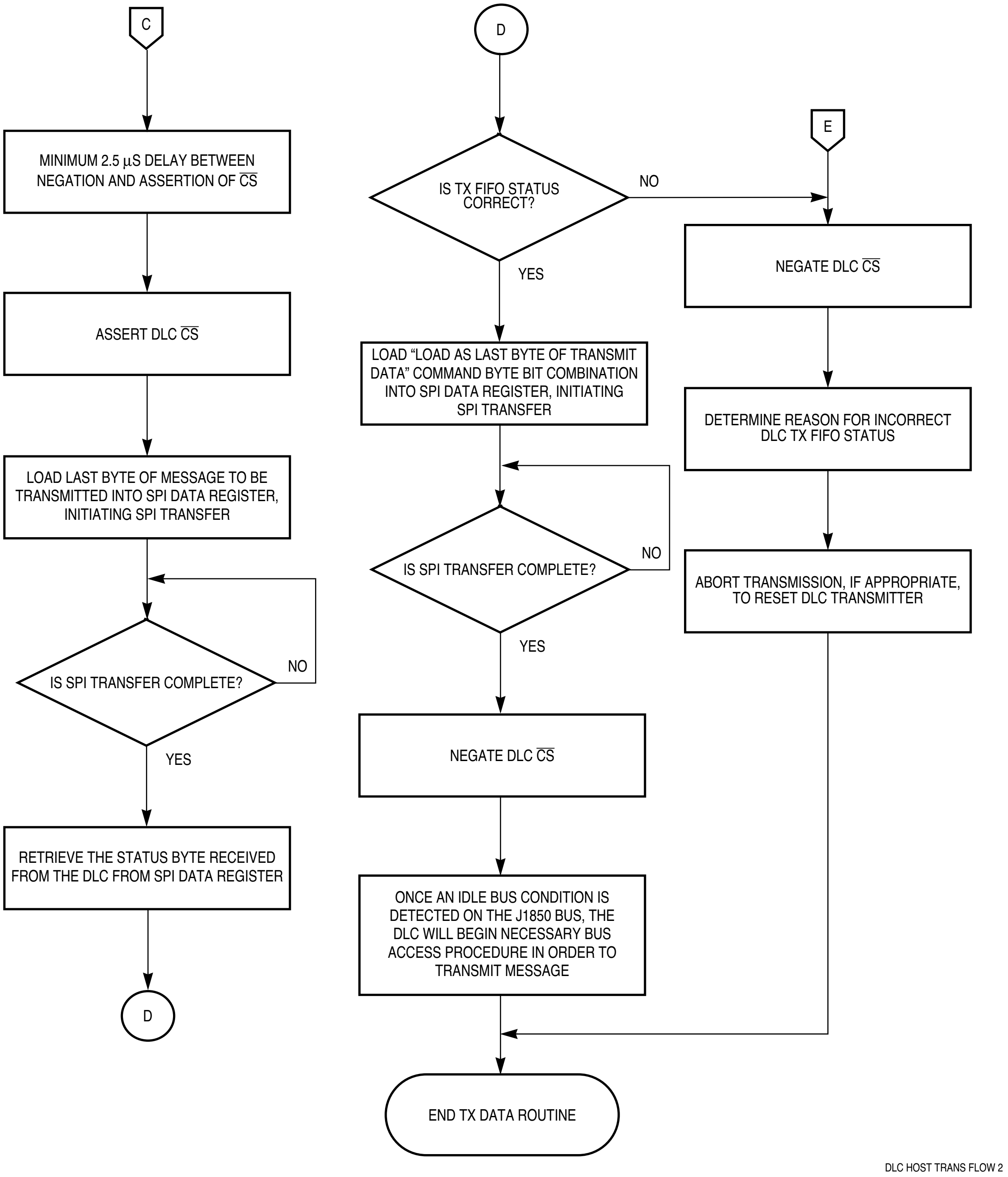

C
MINIMUM 2.5 µS DELAY BETWEEN NEGATION AND ASSERTION OF $\overline{CS}$
ASSERT DLC $\overline{CS}$
LOAD LAST BYTE OF MESSAGE TO BE TRANSMITTED INTO SPI DATA REGISTER, INITIATING SPI TRANSFER
IS SPI TRANSFER COMPLETE?
NO
YES
RETRIEVE THE STATUS BYTE RECEIVED FROM THE DLC FROM SPI DATA REGISTER
D
D
IS TX FIFO STATUS CORRECT?
NO
YES
LOAD "LOAD AS LAST BYTE OF TRANSMIT DATA" COMMAND BYTE BIT COMBINATION INTO SPI DATA REGISTER, INITIATING SPI TRANSFER
IS SPI TRANSFER COMPLETE?
NO
YES
NEGATE DLC $\overline{CS}$
ONCE AN IDLE BUS CONDITION IS DETECTED ON THE J1850 BUS, THE DLC WILL BEGIN NECESSARY BUS ACCESS PROCEDURE IN ORDER TO TRANSMIT MESSAGE
END TX DATA ROUTINE
E
NEGATE DLC $\overline{CS}$
DETERMINE REASON FOR INCORRECT DLC TX FIFO STATUS
ABORT TRANSMISSION, IF APPROPRIATE, TO RESET DLC TRANSMITTER
DLC HOST TRANS FLOW 2

Figure 4-10 Host/DLC Serial Mode Transmit Routine (Part 2 of 2)

4.4 Receiver Operation

The DLC receiver continuously monitors the J1850 bus for voltage swings of ±0.375 volts centered around 3.875 Vdc. It converts these swings to full logic level transitions, which are then clocked through a digital filter into the receiver. Duration of the filtered J1850 bus states are timed by the control logic and compared to a set of received symbol threshold windows. Each J1850 bus state is translated into one of the symbols or is flagged as a timing error. The receiver considers the J1850 bus to be idle when it has been in the passive state for a predetermined time.

Once an SOF symbol is detected, the receiver stores frame bytes as they are received in the RxFIFO buffer until an EOD symbol is detected, or until the RxFIFO buffer is full. As each byte is received, the DLC status is updated to reflect the state of the RxFIFO (the status is updated each clock cycle during the frame).

When all frame bytes have been received, the receiver checks the CRC, but does not place it in the RxFIFO buffer. Instead, it appends a completion code byte to the frame. The byte contains information about the frame.

The DLC receiver performs two basic error detection functions during message transmission and reception. These two error detection mechanisms are:

- The digital input filter
- The receiver J1850 bus monitor

The digital filter eliminates J1850 bus noise spikes and transition noise lasting less than the propagation delay through the J1850 transceiver. Filter operation is best described as a logic-level detector and a 4-bit counter. The counter counts up when the J1850 bus is active (logic level one is detected), and counts down when the J1850 bus is passive (logic level zero is detected). If a full count (0 or 15) occurs, the J1850 bus state transition is considered to be valid, and the signal transition passes to the receiver. As a result, the filter introduces a receive time-delay of 15 to 16 times the internal clock period. With a 2 MHz clock, the clock period is 0.5 µs, so the filter time delay is approximately 8 µs.

Each DLC receives every frame on the J1850 bus, including those it transmits. At the end of each reception from the J1850 bus, the receiver places a completion code byte into the RxFIFO immediately following that frame. This byte contains transmitter action codes, in-frame response codes, and error codes. When a DLC is transmitting, the completion code information applies to both the transmitter and receiver. Completion code bytes are also placed in the RxFIFO when errors are detected, but the host MCU must read and interpret the codes to determine the nature of the error. Available error information includes receiver overrun, transmitter underrun, loss of arbitration, incorrect CRC, incomplete byte indication, and bit timing error. Refer to **SECTION 5 CONTROL AND STATUS CODES** for more information.

A host MCU can monitor the receiver by polling status bytes, or servicing can be interrupt driven. When interrupts are enabled, an interrupt request can be generated by the receiver in the following circumstances:

- When the RxFIFO buffer has received 13 bytes.
- When a completion code is placed in the RxFIFO buffer.
- When a byte has been received into an empty RxFIFO buffer (optional).

For the DLC, status bytes and received data are transferred to the host MCU in pairs. In serial mode, received data remains in the RxFIFO buffer until the host MCU sends a “flush byte” command (data reads and flush commands can be executed at the same time). In parallel mode, the host MCU first reads the status byte, and a subsequent data read automatically flushes the data byte from the RxFIFO buffer.

Figures 4-11 and **4-12** outline the basic software requirements for retrieving data which the DLC in serial mode has received from the J1850 multiplex bus.

NOTE

Particular applications may require more extensive error monitoring and handling routines than this flowchart displays. Conditions reflected in the retrieved status byte (TxFIFO status or J1850 bus status) are also not addressed. Message filtering must be accomplished by the software.

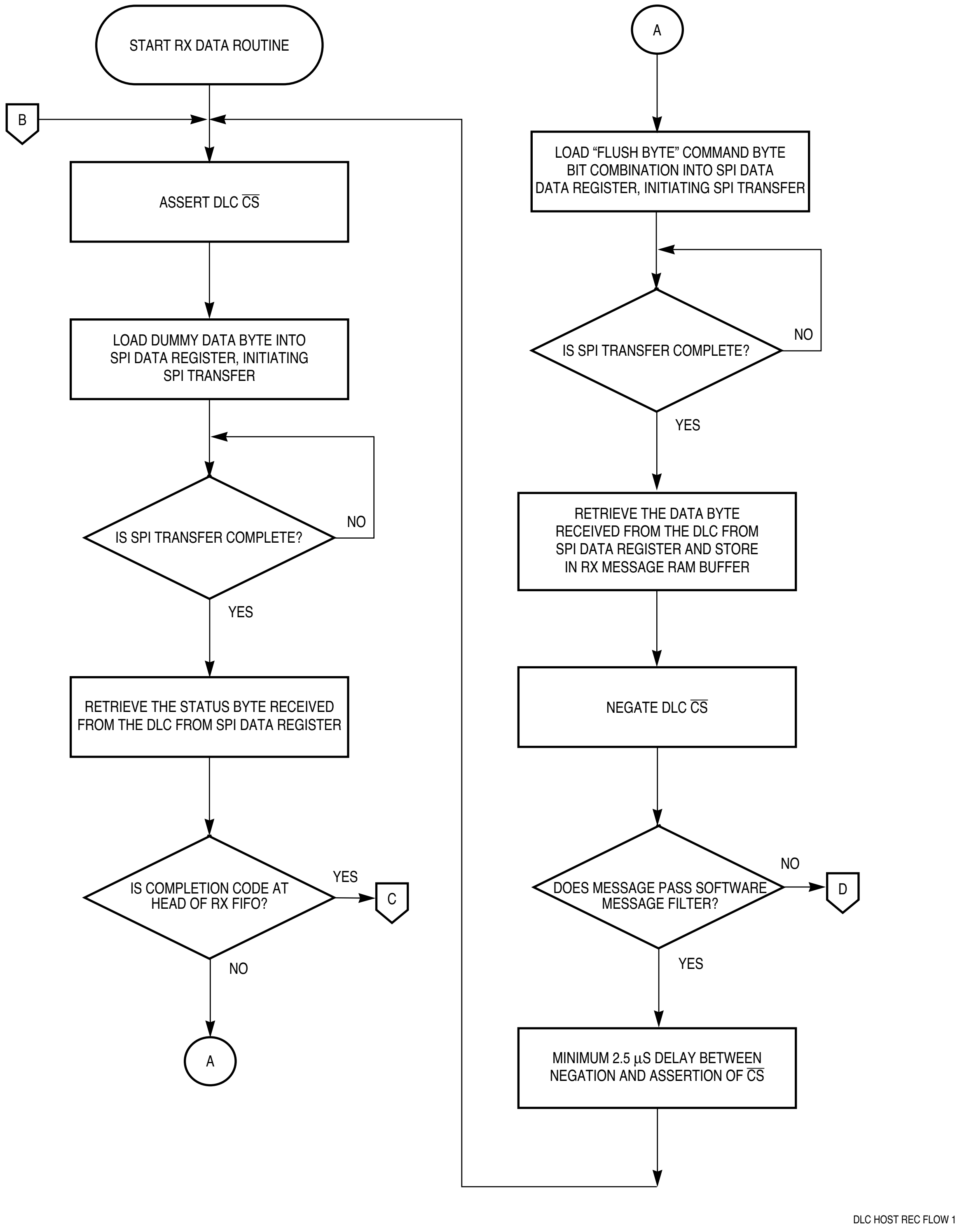

START RX DATA ROUTINE
B
ASSERT DLC $\overline{CS}$
LOAD DUMMY DATA BYTE INTO SPI DATA REGISTER, INITIATING SPI TRANSFER
IS SPI TRANSFER COMPLETE?
NO
YES
RETRIEVE THE STATUS BYTE RECEIVED FROM THE DLC FROM SPI DATA REGISTER
IS COMPLETION CODE AT HEAD OF RX FIFO?
YES
C
NO
A
A
LOAD "FLUSH BYTE" COMMAND BYTE BIT COMBINATION INTO SPI DATA DATA REGISTER, INITIATING SPI TRANSFER
IS SPI TRANSFER COMPLETE?
NO
YES
RETRIEVE THE DATA BYTE RECEIVED FROM THE DLC FROM SPI DATA REGISTER AND STORE IN RX MESSAGE RAM BUFFER
NEGATE DLC $\overline{CS}$
DOES MESSAGE PASS SOFTWARE MESSAGE FILTER?
NO
D
YES
MINIMUM 2.5 µS DELAY BETWEEN NEGATION AND ASSERTION OF $\overline{CS}$
DLC HOST REC FLOW 1

Figure 4-11 Host/DLC Serial Mode Receive Routine (Part 1 of 2)

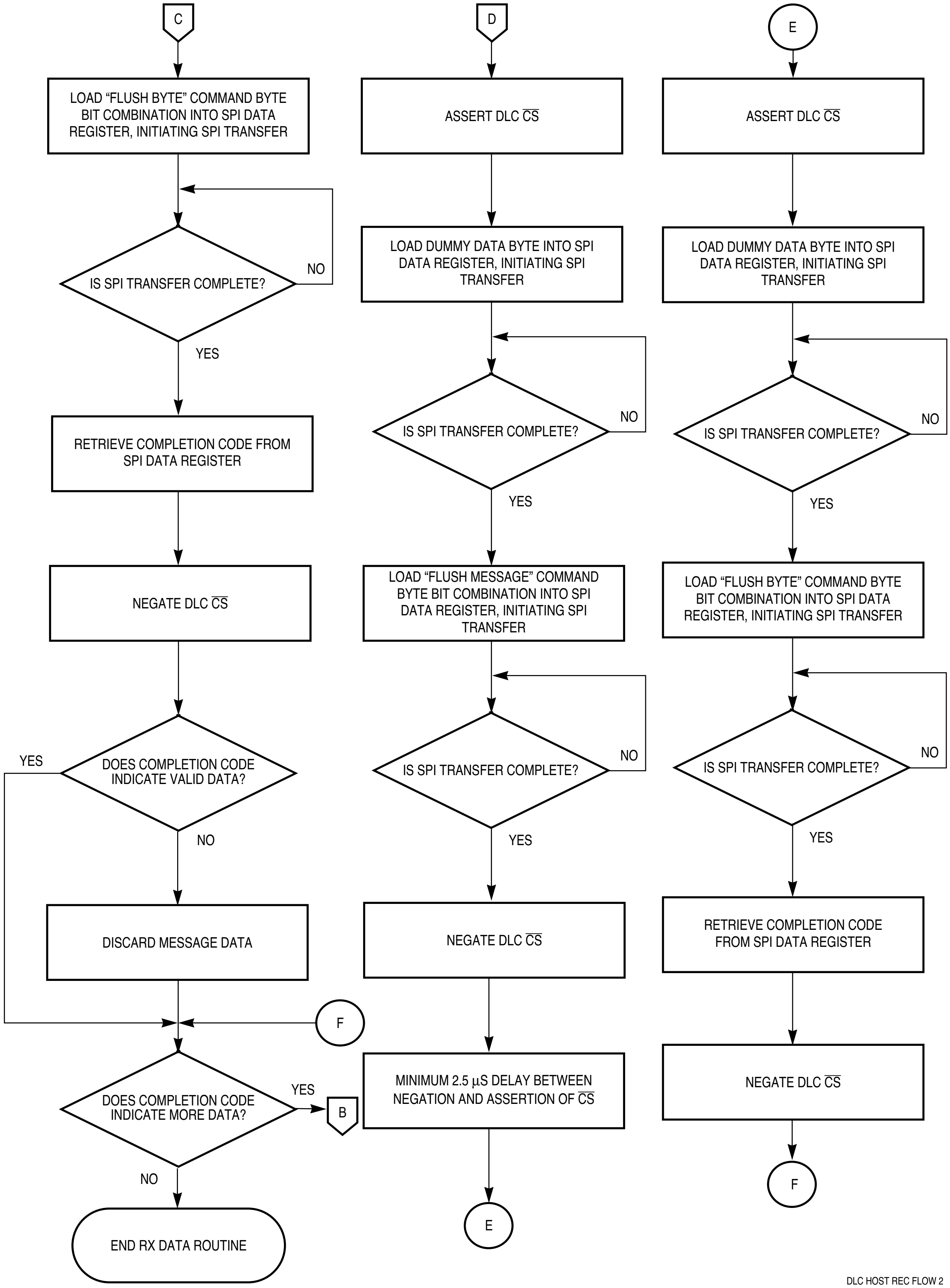

C
LOAD "FLUSH BYTE" COMMAND BYTE BIT COMBINATION INTO SPI DATA REGISTER, INITIATING SPI TRANSFER
IS SPI TRANSFER COMPLETE?
NO
YES
RETRIEVE COMPLETION CODE FROM SPI DATA REGISTER
NEGATE DLC $\overline{CS}$
DOES COMPLETION CODE INDICATE VALID DATA?
YES
NO
DISCARD MESSAGE DATA
F
DOES COMPLETION CODE INDICATE MORE DATA?
YES
B
NO
END RX DATA ROUTINE
D
ASSERT DLC $\overline{CS}$
LOAD DUMMY DATA BYTE INTO SPI DATA REGISTER, INITIATING SPI TRANSFER
IS SPI TRANSFER COMPLETE?
NO
YES
LOAD "FLUSH MESSAGE" COMMAND BYTE BIT COMBINATION INTO SPI DATA REGISTER, INITIATING SPI TRANSFER
IS SPI TRANSFER COMPLETE?
NO
YES
NEGATE DLC $\overline{CS}$
MINIMUM 2.5 µS DELAY BETWEEN NEGATION AND ASSERTION OF $\overline{CS}$
E
E
ASSERT DLC $\overline{CS}$
LOAD DUMMY DATA BYTE INTO SPI DATA REGISTER, INITIATING SPI TRANSFER
IS SPI TRANSFER COMPLETE?
NO
YES
LOAD "FLUSH BYTE" COMMAND BYTE BIT COMBINATION INTO SPI DATA REGISTER, INITIATING SPI TRANSFER
IS SPI TRANSFER COMPLETE?
NO
YES
RETRIEVE COMPLETION CODE FROM SPI DATA REGISTER
NEGATE DLC $\overline{CS}$
F
DLC HOST REC FLOW 2

Figure 4-12 Host/DLC Serial Mode Receive Routine (Part 2 of 2)

4.5 Block Mode Operation

The DLC has the capability of transmitting and receiving message frames which exceed the frame length specified by J1850. These “block mode” messages are typically used in a production or diagnostic environment, and not during normal operation. When operating in block mode, the DLC can transmit or receive message frames containing an unlimited number of data bytes.

The design of the DLC receiver allows it to receive block mode frames into the RxFIFO buffer in a manner identical to normal J1850 frames. As each data byte is received from the J1850 bus, it is placed in the RxFIFO buffer. Throughout reception of the block mode frame, the status byte indicates to the host MCU the status of the RxFIFO buffer. Also, if interrupts are enabled, the DLC generates a host MCU interrupt each time the number of data bytes in the RxFIFO buffer exceeds 12.

As long as the host MCU retrieves the received frame bytes quickly enough to prevent the RxFIFO buffer from overflowing, the DLC continues to place received data bytes into the RxFIFO buffer. Once the EOD symbol is received from the J1850 bus, the CRC byte of the frame is verified, and the appropriate completion code is placed in the RxFIFO buffer. Following this, the status byte indicates that a complete message frame is in the RxFIFO buffer, and a host MCU interrupt, if enabled, is generated. This indicates to the host MCU that the block mode message frame is complete.

If the RxFIFO buffer does overflow, any remaining bytes of the frame are lost, and a completion code is placed in the RxFIFO buffer, indicating that an RxFIFO buffer overrun occurred.

The DLC can transmit a message frame in block mode by completely filling the TxFIFO buffer with data bytes in a manner identical to a normal transmission, except that no byte is accompanied by the “load as last byte of transmit data” command. Once the TxFIFO buffer is completely filled with data bytes, it automatically disables the terminate auto retry feature and begins normal J1850 bus access procedures. As long as the host MCU continues to supply data bytes without latching a “load as last byte of transmit data” command, the DLC continues to transmit the bytes in the block mode frame. When the last data byte of the block mode frame is transferred to the DLC, it should be accompanied by the “load as last byte of transmit data” command. This indicates to the DLC that the end of the block mode frame has been reached.

After the last data byte has been transmitted, the DLC appends the CRC byte to the frame. After the complete frame has been transmitted, the status byte shows the TxFIFO buffer to be empty.

When the transmitter is operating in block mode, the status byte indicates “TxFIFO almost full” when the first frame byte has been sent out onto the J1850 bus, and “TxFIFO contains some data” when the second byte is sent. If a DLC is configured for interrupt operation, an interrupt of the host MCU is generated when the TxFIFO buffer becomes half empty. As the TxFIFO buffer is refilled, status bytes and the interrupt line reflect the changing state of the TxFIFO buffer. Once the “load as last byte of transmit data” command is latched, the status byte shows the TxFIFO buffer to be full until transmission of the block mode message is completed.

If the “load as first byte of transmit data” command is latched while the TxFIFO buffer is full during a block mode transfer, the data byte is not loaded and is lost.

If the “load as first and last byte of transmit data” command is latched while the TxFIFO buffer is full during a block mode transfer, the data byte is not loaded and is lost.

4.6 BREAK Operation

A BREAK symbol on the J1850 bus causes any frame currently on the J1850 bus to be corrupted. The host MCU may command a DLC to send a BREAK symbol. The symbol starts as soon as the command byte is latched.

When a DLC receiver (including any DLC that sends the symbol) detects BREAK on the J1850 bus, it places a completion code in the RxFIFO buffer which indicates that the BREAK was detected. If interrupts are enabled, $\overline{\text{INT}}$ is also asserted. If a frame is being transmitted when the BREAK is detected, the transmitter halts.

Since a BREAK symbol will cause all frames on the J1850 bus to be corrupted, if the DLC is attempting to transmit when the BREAK is detected, it will attempt to retransmit once an IDLE bus condition is detected following the BREAK, unless automatic retry is disabled.

4.7 In-Frame Response (IFR)

The SAE Standard J1850 allows for an optional IFR. A frame on the J1850 bus is determined to be ready for response when an EOD period has elapsed. If a reply to the frame is initiated before the EOF period also elapses, it is designated as an IFR. IFR transmissions are subject to the same arbitration as regular frames, but automatic retry is disabled during an IFR.

If the host wishes to send an IFR, a “send IFR on EOD” command byte accompanies the first byte of an IFR. This command flushes the TxFIFO buffer and queues the IFR.

When the DLC receiver detects an EOD, the IFR is sent, provided the following conditions are met:

- There are no errors in the frame requiring response.
- The IFR is loaded into the TxFIFO buffer after the SOF of the frame requiring response is recognized, but before EOD is detected.
- The first byte of the IFR is accompanied by either a “load as first byte of transmit data” command or “load as first and last byte of transmit data” command.

If these conditions are satisfied, the DLC begins IFR transmission when EOD occurs. If there is no last byte in the TxFIFO buffer when transmission begins, transmission follows normal block-mode rules.

When the “send IFR on EOD” command flushes the TxFIFO buffer, the host MCU must reload and resend frames that are in the TxFIFO buffer when an IFR is queued, even when an IFR is not loaded successfully or loses arbitration.

The IFR bit in the completion code byte indicates whether the associated frame in the RxFIFO buffer is an IFR. When an IFR has been sent, the RxFIFO buffer contains the initial frame, with completion code, plus the IFR and its completion code. The host MCU must check the completion codes to determine that an IFR has been received.

An IFR starts with a normalization (NB) bit. This bit signals whether the IFR was queued by a “send IFR on EOD without CRC” command (NB = 0), or by a “send IFR on EOD with CRC command” (NB = 1). The “in-frame response” command (IFRC) bit in the completion code byte also indicates whether a CRC byte is appended to the IFR. The host MCU must monitor the IFRC if error checking is to be used. Due to zero-dominance, J1850 bus arbitration between an IFR without CRC and an IFR with CRC is won by an IFR without CRC.

NOTE

A DLC cannot transmit an IFR to a message it is transmitting.

Table 4-4 shows IFR error conditions. When errors are detected during an IFR, the IFR is terminated. A completion code indicating that the frame was an IFR and that an error occurred is placed in the RxFIFO buffer.

Table 4-4 IFR Error Conditions

Condition	Action Taken
Error in frame requiring response	Transmitter reset, TxFIFO buffer flushed (no IFR sent)
IFR loses arbitration	Transmission halts, TxFIFO buffer flushed (no IFR sent)
TxFIFO buffer underrun	CRC complemented on truncated IFR
Error in IFR during transmission	Transmission halts, transmitter reset, TxFIFO buffer flushed
J1850 bus idle when IFR loaded	Transmitter reset, TxFIFO buffer flushed
IFR loaded after EOD	Transmitter reset, TxFIFO buffer flushed
IFR loaded without first byte	Transmitter reset, TxFIFO buffer flushed
IFR command, no data	Transmitter reset, TxFIFO buffer flushed

SECTION 5 CONTROL AND STATUS CODES

The host MCU and the DLC exchange control and status information through encoded fields in four types of bytes.

The two byte types containing control information originate in the host MCU. Command bytes direct a DLC to take certain actions. Configuration bytes determine DLC operating parameters.

The two byte types containing status information originate in the DLC. Status bytes convey information concerning DLC buffers and activity to the host MCU. Completion code bytes convey information concerning J1850 bus arbitration, frame type, and errors.

The following paragraphs discuss each type of control and status byte in detail. To use this section, the reader must be familiar with DLC operation and with the interface between the host MCU and the DLC. Refer to **SECTION 4 DATA LINK CONTROLLER OPERATION** for more information on DLC operation. **APPENDIX C DLC REGISTERS** provides a quick reference to all DLC register bit field encodings.

5.1 Command Byte

The values of the fields in a command byte direct the DLC to take specified actions concerning operating mode, configuration bytes, and the TxFIFO and RxFIFO buffers. Commands must be latched into the DLC before they are executed.

CBR — Command Byte Register

	7	6	5	4	3	2	1	0
	GCOM			BTAD			RFC	
RESET:	0	0	0	0	0	0	0	0

5.1.1 GCOM[7:5] — General Command Field

The host MCU directs the DLC to take certain actions by sending it GCOM command codes. **Table 5-1** is a summary of the general commands. Detailed discussion of each command follows.

Table 5-1 General Command Summary

Value	Description
000	Do nothing
001	Enter standby mode
010	Send BREAK symbol
011	Send IFR on EOD with CRC
100	Terminate automatic retry
101	Send IFR on EOD without CRC
110	Reserved
111	Abort transmission

5.1.1.1 Do Nothing

The “do nothing” command indicates to the DLC that a general command should not be executed.

5.1.1.2 Enter Standby Mode

The “enter standby mode” command causes the DLC to operate in low power consumption (standby) mode.

If the J1850 bus is not idle when this command is latched, the DLC becomes inactive within 5 µs after the J1850 bus goes idle.

If the J1850 bus is idle when this command is latched, the DLC goes into standby mode immediately.

Toggling the $\overline{\text{CS}}$ line after this command has been latched, but before the DLC has entered standby mode, clears the command.

5.1.1.3 Send BREAK Symbol

When the “send BREAK symbol” command is latched in, the DLC immediately sends a BREAK symbol on the J1850 bus, regardless of current status of the transmitter.

5.1.1.4 Send IFR on EOD with CRC

The “send IFR on EOD with CRC” command causes the DLC to clear the transmitter and the TxFIFO buffer, terminate automatic frame retry, and originate an IFR with a CRC. This command must accompany a data byte, and the command BTAD field must indicate either “load as first byte of transmit data” or “load as first and last byte of transmit data”. The subsequent bytes in the same IFR must not be accompanied by this command.

If this command is latched after an EOD has already occurred, the IFR is not transmitted, and is lost. When this occurs, the DLC status byte indicates that the TxFIFO buffer is empty, and the DLC does not terminate automatic retry.

If a frame is in progress on the J1850 bus (no EOD has been detected) when this command is latched, the DLC sends the IFR only if no receiver errors are detected. When receiver errors are detected, the IFR is not transmitted, and is lost. If this occurs, the DLC status byte indicates that the TxFIFO buffer is empty, and the DLC does not terminate automatic retry.

If the DLC is transmitting when this command is latched, a timing error or an incomplete byte error will occur. The TxFIFO buffer is flushed, and both the frame and the IFR are lost.

When additional IFR bytes are required, they must be placed in the TxFIFO buffer rapidly enough to prevent underrun. The second IFR byte must be loaded before the falling edge of the Normalization bit, and subsequent bytes must be loaded within 500 µs of each other. If a TxFIFO underrun occurs, it is flagged as an error, and the CRC byte is complemented and appended to the IFR being transmitted.

5.1.1.5 Terminate Auto Retry

The “terminate auto retry” command disables automatic retry. A DLC will normally attempt to retransmit a frame that loses arbitration unless it is sending an IFR or a block mode transmission.

If this command is latched while a transmission is in progress, the DLC attempts to complete the frame. After the frame is successfully transmitted, or after it loses arbitration, automatic retry terminates and the TxFIFO buffer is flushed.

If this command is latched while a complete frame that has lost arbitration is in the buffer awaiting retransmission, the DLC attempts to retransmit the frame once more. After the frame is sent, automatic retry terminates and the TxFIFO buffer is flushed.

If this command is latched while a complete frame that has not been transmitted is in the buffer awaiting transmission, the DLC attempts to send the frame once, then automatic retry terminates and the TxFIFO buffer is flushed.

5.1.1.6 Send IFR on EOD without CRC

The “send IFR on EOD without CRC” command causes a DLC to clear the transmitter and TxFIFO buffer, terminate automatic frame retransmission, and originate an IFR with no CRC. This command must accompany a data byte, and the command byte BTAD field must indicate either “load as first byte of transmit data” or “load as first and last byte of transmit data”. The subsequent bytes in the same IFR must not be accompanied by this command.

If this command is latched while the J1850 bus is idle, or between EOD and EOF, the IFR is not transmitted, and is lost. When this occurs, the DLC status byte indicates that the TxFIFO buffer is empty, and the DLC does not terminate automatic retry.

If a frame is in progress on the J1850 bus (no EOD has been detected) when this command is latched, the DLC sends the IFR only if no receiver errors are detected. When receiver errors are detected, the IFR is not transmitted, and is lost. If this occurs, the DLC status byte indicates that the TxFIFO buffer is empty, and the DLC does not terminate automatic retry.

If the DLC is transmitting when this command is latched, a timing error or an incomplete byte error occurs. The TxFIFO buffer is flushed, and both the frame and the IFR are lost.

When additional IFR bytes are required, they must be placed in the TxFIFO buffer rapidly enough to prevent underrun. The second IFR byte must be loaded before the falling edge of the Normalization bit, and subsequent bytes must be loaded within 500 µs of each other. If a TxFIFO underrun occurs, it is flagged as an error.

5.1.1.7 Abort Transmission

The “abort transmission” command causes the transmitter to be reset immediately. Any transmission in progress is terminated, and the TxFIFO buffer is flushed. The first byte of a new frame can accompany this command.

5.1.2 BTAD[4:2] — Byte Type and Destination Field

The BTAD field tells the DLC how to treat the data byte that accompanies a command byte. **Table 5-2** is a summary of byte type and destination commands. Detailed discussion of each command follows.

Table 5-2 Byte Type and Destination Summary

Value	Description
000	Do not load
001	Load as transmit data
010	Reserved
011	Load as last byte of transmit data
100	Load as configuration byte
101	Load as first byte of transmit data
110	Load as configuration byte-immediate
111	Load as first and last byte of transmit data

5.1.2.1 Do Not Load

The “do not load” command indicates to the DLC that a byte type and destination command should not be executed.

5.1.2.2 Load as Transmit Data

The “load as transmit data” command places an accompanying data byte in the TxFIFO buffer for transmission.

If this command is latched while the TxFIFO buffer is full during a block mode transfer, the data byte is not loaded and is lost.

If this command is latched while a previously loaded complete frame is being transmitted, or is in the TxFIFO buffer awaiting transmission, the data byte is not loaded and is lost.

If this command is latched when the TxFIFO buffer is empty, the data byte is not loaded and is lost.

5.1.2.3 Load as Last Byte of Transmit Data

The “load as last byte of transmit data” command indicates that the accompanying data byte is the last byte of a frame. After the byte is placed in the TxFIFO buffer for transmission, the TxFIFO status field (TMFS) in the status byte is configured to %11 (TxFIFO full).

If this command is latched while a previously loaded complete frame is being transmitted, the accompanying data byte is discarded.

If this command is latched when the TxFIFO buffer is empty, the accompanying data byte is discarded.

5.1.2.4 Load as Configuration Byte

The “load as configuration byte” command indicates that the accompanying data byte contains DLC configuration data. Changes in configuration are performed after current J1850 bus activity is finished unless this command immediately follows reset. In this case, the DLC is configured at least 1.2 ms after the command is latched.

5.1.2.5 Load as First Byte of Transmit Data

The “load as first byte of transmit data” command indicates that the accompanying data byte is the first byte of a frame. After the byte is placed in the TxFIFO buffer for transmission, the TMFS in the status byte is set to %01 (TxFIFO contains data byte).

If the transmitter encounters another “first byte” command in the buffer before decoding a valid “last byte” command, it sends a complemented CRC and halts. A completion code with the error flag (ERRF) bit set and a transmitter status (TMS) field value of %10 (transmitter underrun) is placed in the RxFIFO buffer. The second “first byte” command and any succeeding bytes are not transmitted.

If this command is latched while a previously loaded complete frame is being transmitted, the data byte is not loaded and is lost.

5.1.2.6 Load as Configuration Byte – Immediate

The “load as configuration byte – immediate” command indicates that the accompanying data byte contains DLC configuration data. Changes in configuration are performed immediately. This command should only be used for system initialization and test.

5.1.2.7 Load as First and Last Byte of Frame

The “load as first and last byte of frame” command indicates that the accompanying data byte is a one byte frame. After the byte is placed in the TxFIFO buffer for transmission, the TMFS in the status byte is set to %11 (TxFIFO full).

If this command is latched while a previously loaded complete frame is in the buffer awaiting transmission, the accompanying data byte will be discarded.

If this command is latched while a previously loaded complete frame is being transmitted, the accompanying data byte will be discarded.

5.1.3 RFC[1:0] — Receive FIFO Command Field

Table 5-3 displays the RxFIFO commands.

Table 5-3 RFC Field Encoding

Value	Description
00	Do nothing
01	Reserved
10	Flush byte or completion code
11	Flush frame except for completion code

5.1.3.1 Do Nothing

The “do nothing” command indicates to the DLC that an RxFIFO buffer command should not be executed.

5.1.3.2 Flush Byte

The “flush byte” command flushes the first data byte or completion code in the RxFIFO buffer. “Flush byte” commands cannot be queued. When there is nothing in the buffer, no action is taken if a byte arrives after the command is latched. This is the only command that can flush a completion code.

5.1.3.3 Flush Frame

The “flush frame” command flushes the current frame except for the completion code generated when the frame has been completely received. If interrupts are enabled when a “flush frame” command is executed, the receiver will only generate an interrupt request once the completion code is at the head of the RxFIFO.

When the RxFIFO buffer is flushed, it is not completely cleared, but cycled until the first byte in the RxFIFO buffer is a completion code.

If the command is latched when a complete frame is in the buffer, the entire frame, except for the completion code, is flushed.

If the command is latched when there is nothing in the buffer, the next frame, except for the completion code, is flushed.

If the command is latched while a frame is being received, all bytes in the RxFIFO buffer and all subsequent bytes received before the completion code is generated, are flushed.

In the two previous instances, the RxFIFO status (RFS) field in the status byte may be invalid until the completion code is recognized.

If the command is latched while the first byte in the RxFIFO buffer is a completion code, the RxFIFO buffer is not flushed.

5.2 Configuration Byte

The fields in the configuration byte determine the basic operational parameters of the DLC.

A host MCU cannot read configuration information from the DLC. Once a configuration byte is sent to the DLC, operating parameters cannot change until a new byte is sent, or a reset occurs.

CBR — Configuration Byte Register

	7	6	5	4	3	2	1	0
	TM	TC		IMSK	IMOD	OSCD		4X
RESET:	0	0	0	0	0	1	1	0

5.2.1 TM — Test Mode Control Bit

Setting the TM bit enables the receive data disable test mode. This mode is used for Motorola internal testing only.

0 = Normal operation
1 = Test mode

5.2.2 TC[6:5] — Test Configuration Field

The TC bit field selects one of four operating modes. When TC value is %00 (default), the DLC operates normally. Each of the other values selects a test mode. These modes are used for Motorola internal testing only.

5.2.3 IMSK — Interrupt Mask Bit

The IMSK bit determines DLC interrupt request capability. When interrupts are disabled, the DLC cannot exit standby mode upon detection of J1850 bus activity.

0 = All interrupts to the MCU are enabled
1 = All interrupts to the MCU are disabled

5.2.4 IMOD — Interrupt Mode Bit

The IMOD bit controls an additional interrupt source to the default interrupt sources. When IMOD is set, an interrupt request is made when a data byte is received into an empty RxFIFO buffer. Subsequent bytes do not cause an interrupt request unless the RxFIFO is emptied before they are received.

0 = Default interrupts are the only ones enabled
1 = Additional interrupt source added to default sources

5.2.5 OSCD[2:1] — Oscillator Divisor Field

The OSCD bit field determines DLC internal clock frequency. The DLC internal clock frequency is derived from a combination of the external clock frequency and the OSCD value. **Table 5-4** shows frequency division factors and the internal clock frequency with various references. A 2 MHz internal clock frequency is required for normal operation.

Table 5-4 Internal Clock Frequency Derivations

OSCD	Clock	External Clock			
Value	Divisor	2 MHz	4 MHz	6 MHz	8 MHz
00	1	2 MHz	4 MHz	6 MHz	8 MHz
01	2	1 MHz	2 MHz	3 MHz	4 MHz
10	3	0.66 MHz	1.33 MHz	2 MHz	2.66 MHz
11	4	500 kHz	1 MHz	1.5 MHz	2 MHz

5.2.6 4X — High-Speed Control Bit

Setting this bit places the DLC in high-speed data transfer mode. J1850 bus waveshaping is disabled.

0 = Normal clock division
1 = Four times normal clock speed

5.3 Status Byte

The status byte register conveys information about data and shows the condition of the DLC at the time of receipt. Results of commands that are in progress may not be reflected in the status byte until the command is completely executed.

For a DLC operating in parallel mode, the host MCU can read the status byte alone, and a subsequent data read automatically flushes the data byte from the buffer.

For a DLC operating in SPI mode, status bytes and received data are transferred to the host MCU in pairs. When the host reads received data, it remains in the RxFIFO buffer until a flush command is given. The flush command can accompany the read command. The DLC does not report the status of an action caused by the current transfer.

All information in the status byte is frozen while $\overline{CS}$ is asserted, and updated when $\overline{CS}$ is negated.

SBR — Status Byte Register

	7	6	5	4	3	2	1	0
	RFS			DLI	NETF	4XMD	TMFS	
RESET:	0	0	0	X	0	0	0	0

5.3.1 RFS[7:5] — Receive FIFO Status Field

This bit field shows the status of the RxFIFO buffer. When interrupts are enabled, RFS can be used to determine the source of an interrupt request. **Table 5-5** shows the RxFIFO status encoding field. In the DLC, when the RFS bit field indicates that a completion code is at the head of the buffer, the accompanying data byte contains that same completion code.

Table 5-5 RFS Field Encoding

Value	Description
000	Buffer invalid or empty
001	Buffer contains more than one byte
010	Buffer contains a completion code
011	Data byte in 13th buffer position, no completion code
100	One data byte in buffer
101	Completion code at head of buffer, more bytes available
110	Completion code at head of buffer, another frame available
111	Completion code only at head of buffer

5.3.1.1 Buffer Invalid or Empty

This value indicates that the accompanying data byte is not valid because the RxFIFO buffer is empty or a flush frame command is being executed.

5.3.1.2 Buffer Contains More Than One Byte

This value indicates that the RxFIFO buffer contains two to 12 bytes of data and no completion code.

5.3.1.3 Buffer Contains a Completion Code

This value indicates that the RxFIFO buffer contains both data bytes and a completion code. The completion code is not at the head of the buffer.

5.3.1.4 Thirteenth Byte Received

This value indicates that the RxFIFO buffer contains 12 bytes and a 13th byte has been received. There is no completion code in the buffer.

5.3.1.5 One Byte in Buffer

This value indicates that the RxFIFO buffer contains one byte that is not a completion code.

5.3.1.6 Completion Code at Head of Buffer, More Bytes Available

This value indicates that there is a completion code at the head of the RxFIFO buffer, and additional bytes but no additional completion codes are in the buffer.

5.3.1.7 Completion Code at Head of Buffer, Frame Available

This value indicates that there is a completion code at the head of the RxFIFO buffer, and additional bytes and an additional completion code are in the buffer.

5.3.1.8 Completion Code Only at Head of Buffer

This value indicates that there is a completion code at the head of the RxFIFO buffer, and no additional bytes are in the buffer. Indication usually occurs after execution of a flush frame command.

5.3.2 DLI — Data Link Idle Bit

DLI is set when the J1850 bus is idle. DLI is cleared when the J1850 bus has been active for longer than the digital filter time constant, or when an EOD symbol is being timed by the receive logic. Once cleared, DLI does not go to logic level one until an EOD symbol has been received. If the J1850 bus remains active for longer than the minimum BREAK symbol definition, a completion code indicating a break has occurred is placed in the RxFIFO buffer. A prolonged data link active indication may indicate that the J1850 bus is shorted to 12 Vdc.

0 = SAE J1850 bus is active
1 = SAE J1850 bus is idle

5.3.3 NETF — Network Fault Bit

This value indicates the DLC J1850 bus connection is shorted to ground. When a hardware fault prevents the DLC receiver from sensing an active state after the transmitter has driven the J1850 bus for 60 µs, NETF is set. After NETF is set, only an Abort Transmission command or assertion of the DLC $\overline{\text{RST}}$ input can restart the DLC.

When the short is momentary, it is appropriate to try reloading and retransmission. However, if the problem is a defective receiver, retransmission could corrupt another frame.

0 = Active state occurred before 60 µs
1 = Active state not sensed after 60 µs

5.3.4 4XMD — 4X Mode Bit

This value indicates the DLC is operating in high-speed mode without waveshaping. This mode is not used during normal operation.

0 = DLC operating in normal mode
1 = DLC operating at 41.67 kbps

5.3.5 TMFS[1:0] — TxFIFO Status Field

This bit field shows the status of TxFIFO buffer. **Table 5-6** shows the TxFIFO status encoding field.

Table 5-6 TMFS Field Encoding

Value	Description
00	Buffer empty
01	Buffer contains data bytes
10	Buffer almost full
11	Buffer full

5.3.5.1 Buffer Empty

This value indicates there are no data bytes in TxFIFO buffer.

5.3.5.2 Buffer Contains Data

This value indicates that the TxFIFO buffer contains one to nine bytes of data.

5.3.5.3 Buffer Almost Full

This value indicates that the TxFIFO buffer contains ten bytes of data.

5.3.5.4 Buffer Full

This value indicates that either the TxFIFO buffer contains 11 bytes of data (in block transmission mode), or that it contains a byte that was accompanied by a “last byte” command.

5.4 Completion Code Byte

When a frame is complete, after the time period for an EOD symbol has elapsed, the DLC generates a completion code byte and transfers it to the host MCU in the same way that a received data byte is transferred. The accompanying status byte indicates that it is a completion code byte.

The completion code byte contains an error flag, transmitter action codes, IFR codes, and error codes. When a DLC is transmitting, completion code information applies to both transmitter and receiver.

Any activity on the J1850 bus longer than the digital filter time constant causes a completion code byte to be placed in the RxFIFO buffer after the J1850 bus is idle for an EOD length of time.

CCBR — Completion Code Byte Register

	7	6	5	4	3	2	1	0
	ERRF	RFO	TMS		IFR	IFRC	ERRC	
RESET	0	0	0	0	0	0	0	0

5.4.1 ERRF — Error Bit

This bit indicates whether the receiver detected an error during the frame. When ERRF is set, the ERRC field indicates which error occurred.

0 = No error occurred
1 = Error occurred

5.4.2 RFO — Receive FIFO Overrun Bit

When the RFO bit is set, a RxFIFO buffer overrun has occurred. This means that the host MCU has not read the RxFIFO buffer regularly enough to prevent incoming frame bytes from being lost. RFO is set when the first bit of the 21st data byte is received. Previously received data bytes remain in the buffer. When RFO is set, no other completion code bits can be considered valid. There is no way to detect BREAK when the buffer has overflowed.

0 = No receiver buffer overrun occurred
1 = Receiver buffer overrun occurred

5.4.3 TMS[5:4] — Transmitter Status Field

This bit field shows transmitter status during the frame just received. **Table 5-7** shows transmitter status field encoding.

Table 5-7 TMS Field Encoding

Value	Description
00	Transmitter not involved
01	Transmitter underrun
10	Transmitter lost arbitration
11	Transmitter successful

5.4.3.1 Transmitter Not Involved

This value indicates that the transmitter did not perform J1850 bus contention for this frame. Frame did not originate from this node.

5.4.3.2 Transmitter Underrun

This value indicates that a frame was transmitted but was not completed. This is not an indication that the transmission lost J1850 bus contention.

5.4.3.3 Transmitter Lost Arbitration

This value indicates that the transmitter lost contention, or an RxFIFO buffer overrun occurred. If overrun occurred, transmitter may not have been arbitrated against, and transmission may not have stopped.

5.4.3.4 Transmitter Successful

This value indicates that a frame transmitted successfully. Frame originated from this node.

5.4.4 IFR — In-Frame Response Bit

This bit indicates whether or not a frame is an IFR.

0 = Not an IFR
1 = IFR

5.4.5 IFRC — In-Frame Response CRC Bit

This bit indicates whether or not an IFR has an appended cyclic redundancy check.

0 = IFR without CRC
1 = IFR with CRC

5.4.6 ERRC[1:0] — Error Code Field

This bit field is used in conjunction with the error flag. It indicates which errors were detected. Errors are reported in precedence order; the higher the ERRC value, the higher the priority. If any of these error conditions are detected, a pending EOD response frame (IFR) is not sent. **Table 5-8** shows the ERRC encoding field.

Table 5-8 ERRC Field Encoding

Value	Description
00	CRC error
01	Incomplete byte
10	Bit timing error
11	Break

5.4.6.1 CRC Error

This value indicates that the cyclic redundancy check is incorrect or that a receiver overrun has occurred.

5.4.6.2 Incomplete Byte Error

This value indicates that an incorrect modulus eight count occurred during reception of this frame, and that an incomplete byte was received.

5.4.6.3 Bit Timing Error

This value indicates that a bit in the received frame did not match the timing window stored in the control logic. The receiver immediately places a completion code byte in the buffer upon detection of this error. This generally occurs as a result of mismatched clock divisors between nodes, but may occur if a BREAK is detected after the J1850 bus has been passive for 8 to 34 µs. When this occurs, two completion codes (a bit error and a break) are generated.

5.4.6.4 BREAK Error

This value indicates that a BREAK symbol was detected on the J1850 bus, or the J1850 bus was shorted high for longer than the defined BREAK period, and has recovered.

APPENDIX A ELECTRICAL CHARACTERISTICS

Table A-1 Operating Conditions

Symbol	Definition	Conditions	Min	Typ	Max	Units
V_{BATT}	Voltage at battery pin during transmit operation	Rs = 20 Ω max. (protection circuit)	7.0	12.0	26.5	V
	Voltage at battery pin during receive operation	Rs = 20 Ω max. (protection circuit)	4.9	12.0	26.5	V
V_{CC}, V_{DD}	Transceiver, logic supplies	—	4.75	5.0	5.25	V
PC_{MAX}	Maximum continuous power dissipation	7.8 kHz bus waveform (50% duty cycle); 4.00 MHz resonator; 16-volt battery	—	—	333	mW
I_{BATT}, standby	Battery standby current	DLC in standby mode	—	3	5	µA
I_{BATT}, draw	Battery current draw	7.8 kHz bus waveform (50% duty cycle); 4.00 MHz resonator; 16-volt battery	—	3.5	50	mA
I_{DD}, draw	Logic current draw	100% bus utilization 4.00 MHz resonator	—	1.5	4	mA
I_{DD}, draw, slp[1]	Logic current draw, standby mode	DLC in sleep mode	—	3	5	µA
I_{CC}, draw	Transceiver current draw	100% bus utilization 4.00 MHz resonator	—	0.3	1	mA
I_{CC}, draw, slp	Transceiver current draw, standby mode	DLC in sleep mode	—	40	55	µA
I_{PSEN}, source	PSEN pin max source	V_{BATT} = 9 V Rseries = 33 kΩ	10	—	500	µA
I_{OLPSEN}	PSEN leakage	Standby mode, power supply off	–5	—	5	µA
V_{PSEN}	PSEN pin output voltage	V_{BATT} = 9 V Rseries = 33 kΩ	8.3	—	—	V
V_{LH}	Input range on all logic pins	—	–0.5	—	5.75	V
T_{ST}	Storage temperature range	—	–65	—	150	°C
T_J	Maximum junction temperature	—	—	—	150	°C
T_A	Operating temperature range	—	–40	—	125	°C
I_{BAT}, sleep	Battery sleep current	DLC in sleep mode	—	—	5	mA

NOTES:

1. Measured with no activity on the host /DLC interface lines.

Table A-2 Electrical Characteristics

Symbol	Description	Conditions	Min	Max	Units
V_{OH}[1]	Minimum guaranteed output high voltage	I_{OH}= –200 µA	V_{DD} – 0.8V	—	V
V_{OL}	Maximum guaranteed output low voltage	I_{OL} = 1.6 mA	—	0.4	V
V_{IH}	Minimum guaranteed input high voltage	I_{IN} = 10 µA	V_{DD} x 0.7V	—	V
V_{IL}	Maximum guaranteed input low voltage	I_{IN} = 10 µA	—	0.8	V
I_{IN} (except OSC1, $\overline{CS}$)	Input current limits	—	–10	10	µA
$I_{IH, OSC1}$	Maximum guaranteed input high current	V_{DD} = 4.75 to 5.25 V	0.525	10.50	µA
$I_{IL, OSC1}$	Maximum guaranteed input low current	V_{DD} = 4.75 to 5.25 V	–10.50	–0.525	µA
$I_{IN\overline{CS}}$	Chip select input current limit	—	–650	50	µA
I_{OZ} (except OSC1)	Output leakage high-Z	V_{OUT} to GND	–10	10	µA
C_I	Digital input capacitance	—	—	8	pF
C_{IO}	Digital input/output capacitance	—	—	10	pF
C_{LOAD}	Maximum load capacitance	—	—	100	pF

NOTES:

1. Interrupt request line is an open drain output; therefore, V_{OH} is not applicable.

Table A-3 Absolute Maximum Ratings

Symbol	Description	Min	Max	Units
V_{BATT}	Supply voltage	–0.5	26.5	V
V_{BATT}	Supply voltage (t ≤ 1 ms)	—	40.0	V
V_{CC},V_{DD}	Supply voltage	–0.5	5.75	V
T_{STG}	Storage temperature range	–65	+150.0	°C
T_J	Maximum junction temperature	—	+150.0	°C

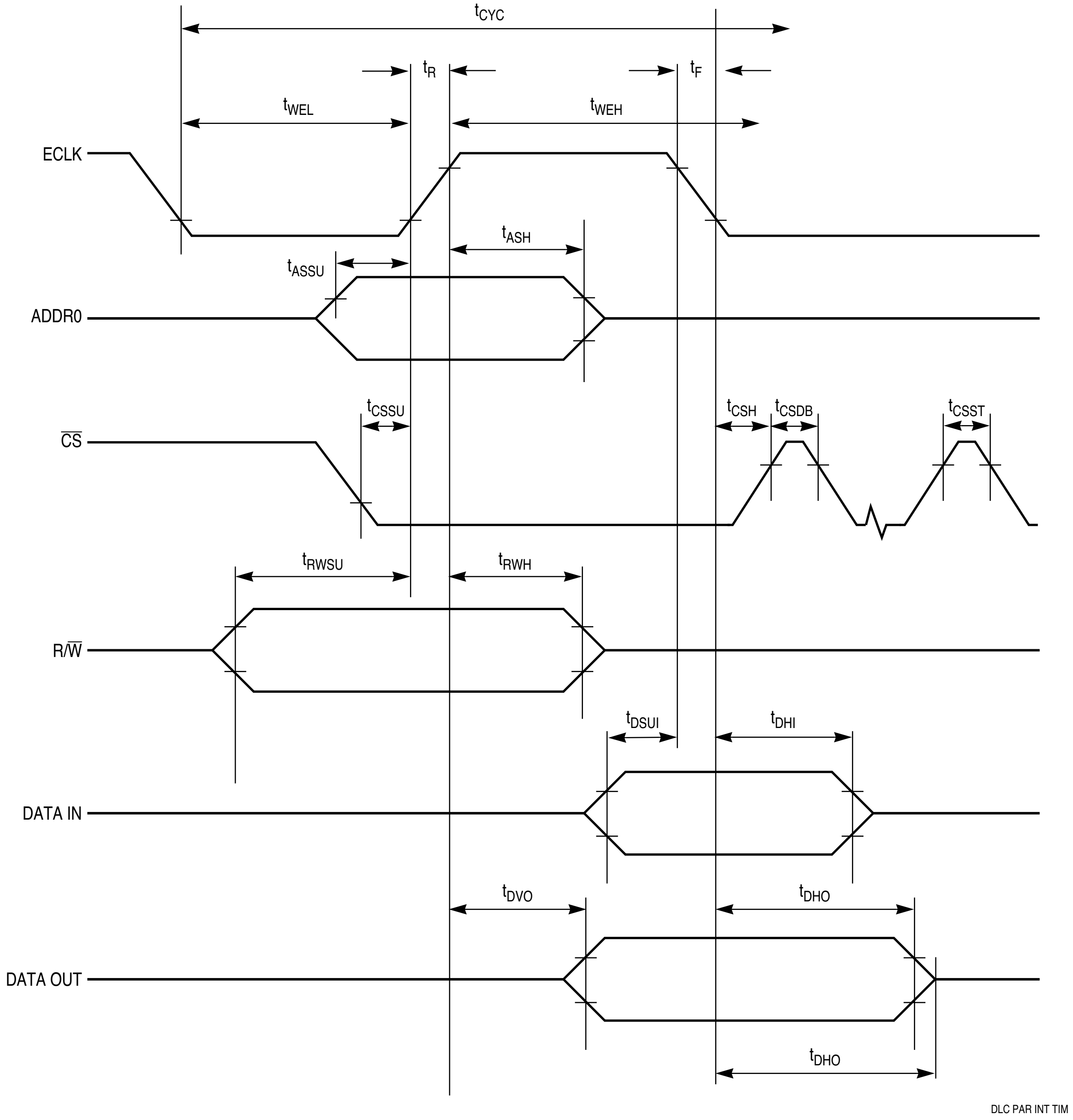


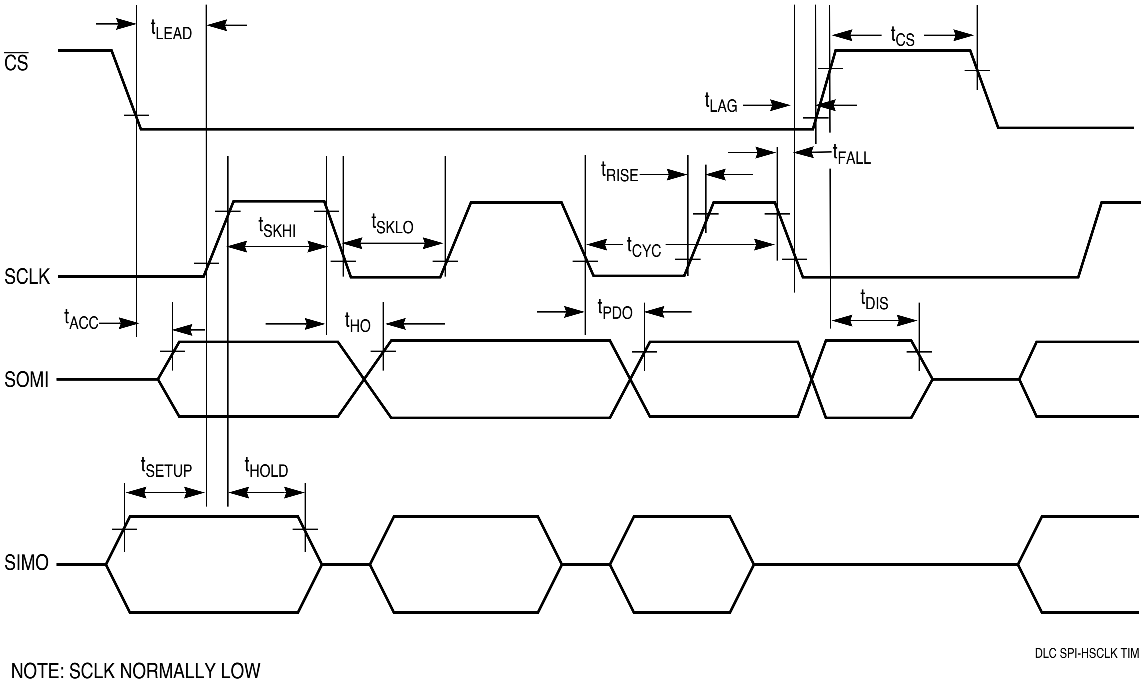
Figure A-1 Parallel Interface Timing

Table A-4 Parallel Interface Parameters

Symbol	Description	Pin	Min	Max	Units
t_{CYC}	E-clock cycle time	EC	238	—	ns
t_{WEL}	E-clock pulse width low	EC	105	—	ns
t_{WEH}	E-clock pulse width high	EC	100	—	ns
$t_{R,F}$	E-clock rise and fall time	EC	0	20	ns
t_{ASSU}	Address select setup time	A0	20	—	ns
t_{ASH}	Address select hold time	A0	20	—	ns
t_{CSSU}	Chip-select setup time	$\overline{CS}$	5	—	ns
t_{CSH}	Chip-select hold time	$\overline{CS}$	5	—	ns
t_{CSDB}	Chip-select double[1]	$\overline{CS}$	0	60	ns
t_{CSST}	Chip-select status time[2, 3]	$\overline{CS}$	3	—	µs
t_{RWSU}	R/$\overline{W}$ setup time	R/$\overline{W}$	25	—	ns
t_{RWH}	R/$\overline{W}$ hold time	R/$\overline{W}$	20	—	ns
t_{DSUI}	Data in setup time	D0–D7	45	—	ns
t_{DHI}	Data in hold time	D0–D7	15	—	ns
t_{DVO}	Data out valid time	D0–D7	—	75	ns
t_{DHO}	Data out hold time	D0–D7	5	—	ns
t_{DTO}	Data out three-state time	D0–D7	—	40	ns

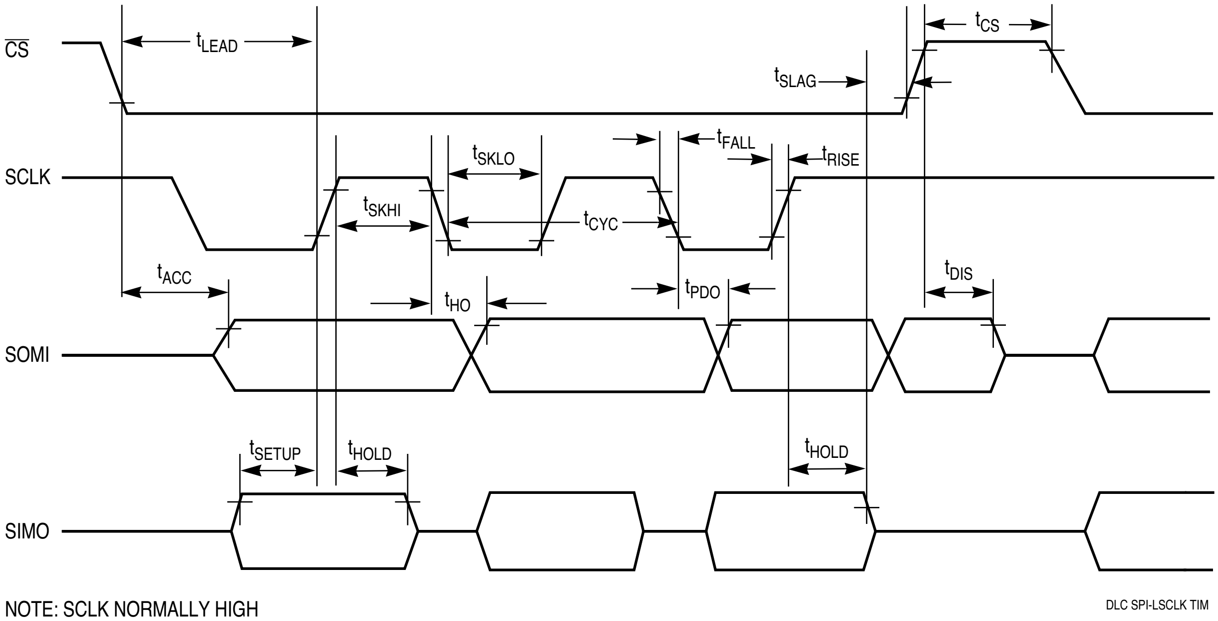
NOTES:

1. Within double-byte reads and writes.
2. Between successive commands; also between consecutive “auto-finishes” of data in DLC parallel mode.
3. Needed between successive status byte reads to properly update status.



NOTE: SCLK NORMALLY LOW

Figure A-2 SPI Timing — Active High SCLK



NOTE: SCLK NORMALLY HIGH

Figure A-3 SPI Timing — Active Low SCLK

Table A-5 Serial Interface Parameters

Symbol	Description	Min	Max	Units
t_{CS}	Minimum time between consecutive $\overline{CS}$ assertions	3.0	—	µs
t_{CYC}	Minimum SCK cycle time[1]	238	—	ns
t_{SKHI}	Minimum clock high time	80	—	ns
t_{SKLO}	Minimum clock low time	80	—	ns
t_{LEAD}	Minimum enable lead time	100	—	ns
t_{LAG}	Minimum enable lag time	100	—	ns
t_{ACC}	Access time	—	60	ns
t_{PDO}	Maximum data out delay time	—	59	ns
t_{HO}	Minimum data out hold time	0	—	ns
t_{DIS}	Maximum data out disable time	—	240	ns
t_{SETUP}	Minimum data setup time	30	—	ns
t_{HOLD}	Minimum data hold time	30	—	ns
t_{RISE}	Maximum time for input to go from V_{OL} to V_{OH}	—	25	ns
t_{FALL}	Maximum time for input to go from V_{OH} to V_{OL}	—	25	ns
t_{SLAG}	Minimum time after data hold time that $\overline{CS}$ may be negated	100	—	ns

NOTES:

1. Maximum SPI frequency is 4.2 MHz.

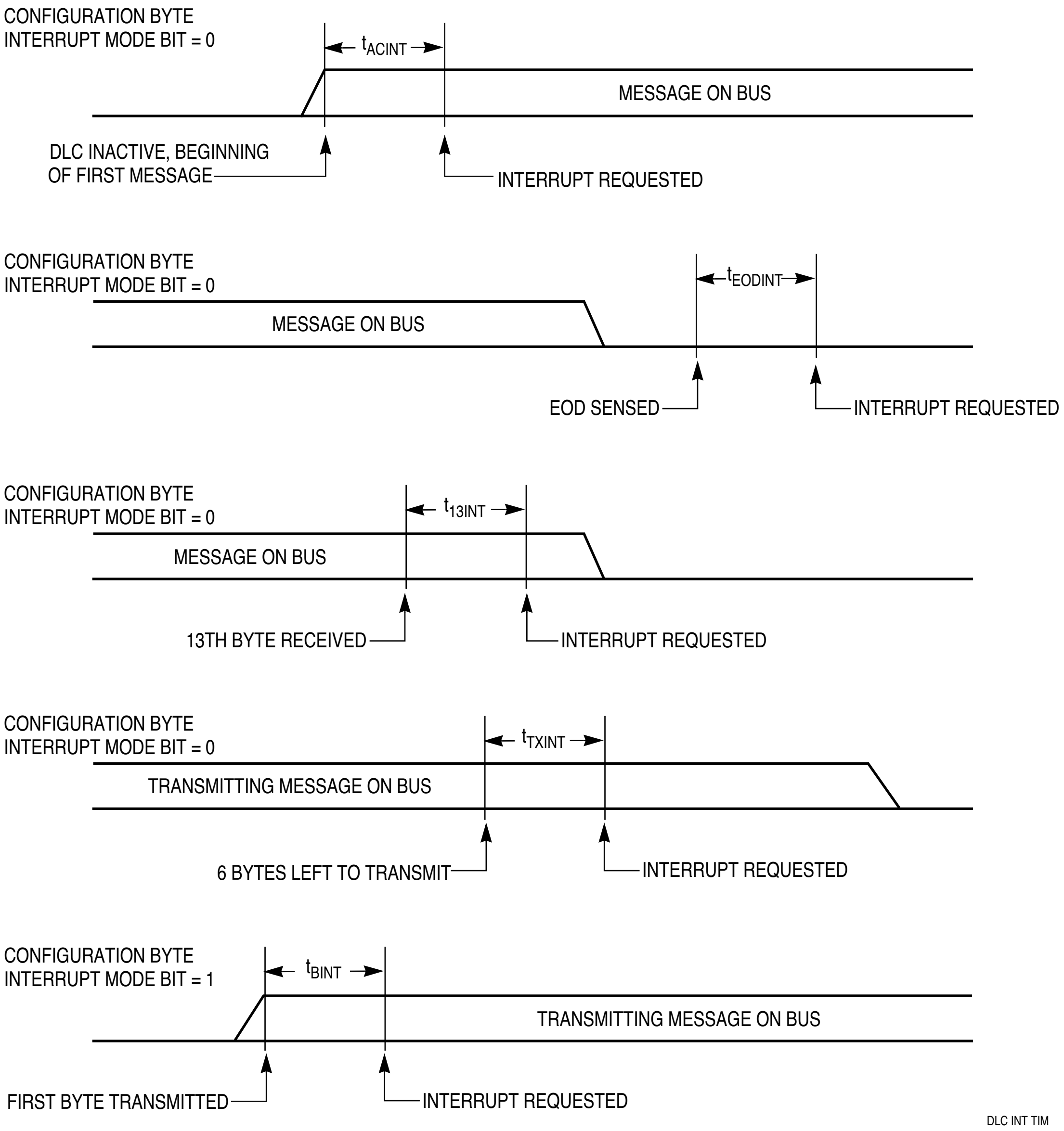


Figure A-4 DLC Interrupt Timing

Table A-6 Standby and Interrupt Timing

Symbol	Description	Min	Max	Units
t_{ACDLY}	Period from detection of bus activity until assertion of PSEN, or period from application of 5 Vdc until assertion of PSEN	—	5	µs
t_{ACINT}	Period from detection of bus activity by a DLC in standby condition until $\overline{\text{INT}}$ assertion	—	105	µs
t_{EODINT}	Period from detection of EOD on the bus until $\overline{\text{INT}}$ assertion	—	5	µs
t_{13INT}	Period from receipt of 13th byte until $\overline{\text{INT}}$ assertion	—	5	µs
t_{TXINT}	Period from when there are six bytes left to transmit until $\overline{\text{INT}}$ assertion	—	5	µs
t_{BINT}	Period from receipt of first byte until $\overline{\text{INT}}$ assertion	—	5	µs

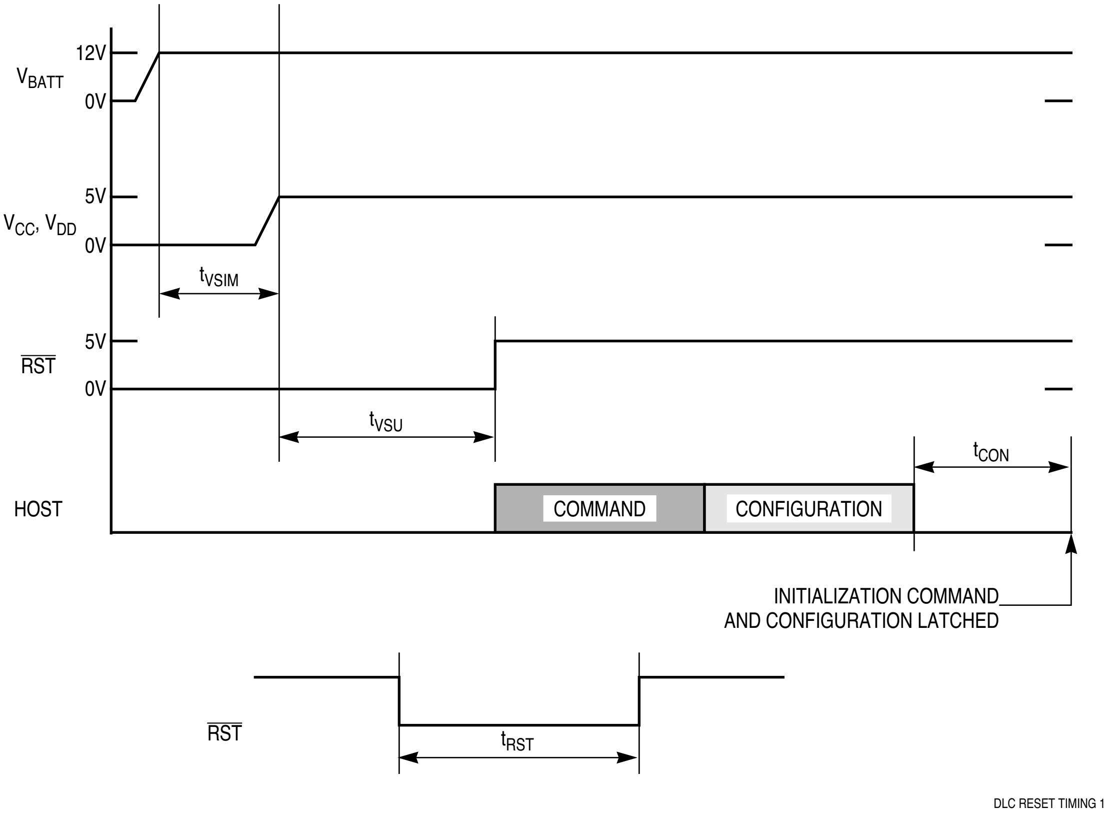


Figure A-5 Reset Timing

Table A-7 Reset Timing

Symbol	Description	Min	Max	Units
t_{VSIM}	V_{DD}, V_{SS} simultaneous switch delay	6	10	ms
t_{VSU}	V_{DD}, V_{SS} set up time	50	100	ms
t_{CON}	Immediate configuration time	300	1200	µs
t_{RST}	Reset pulse width	1	—	µs

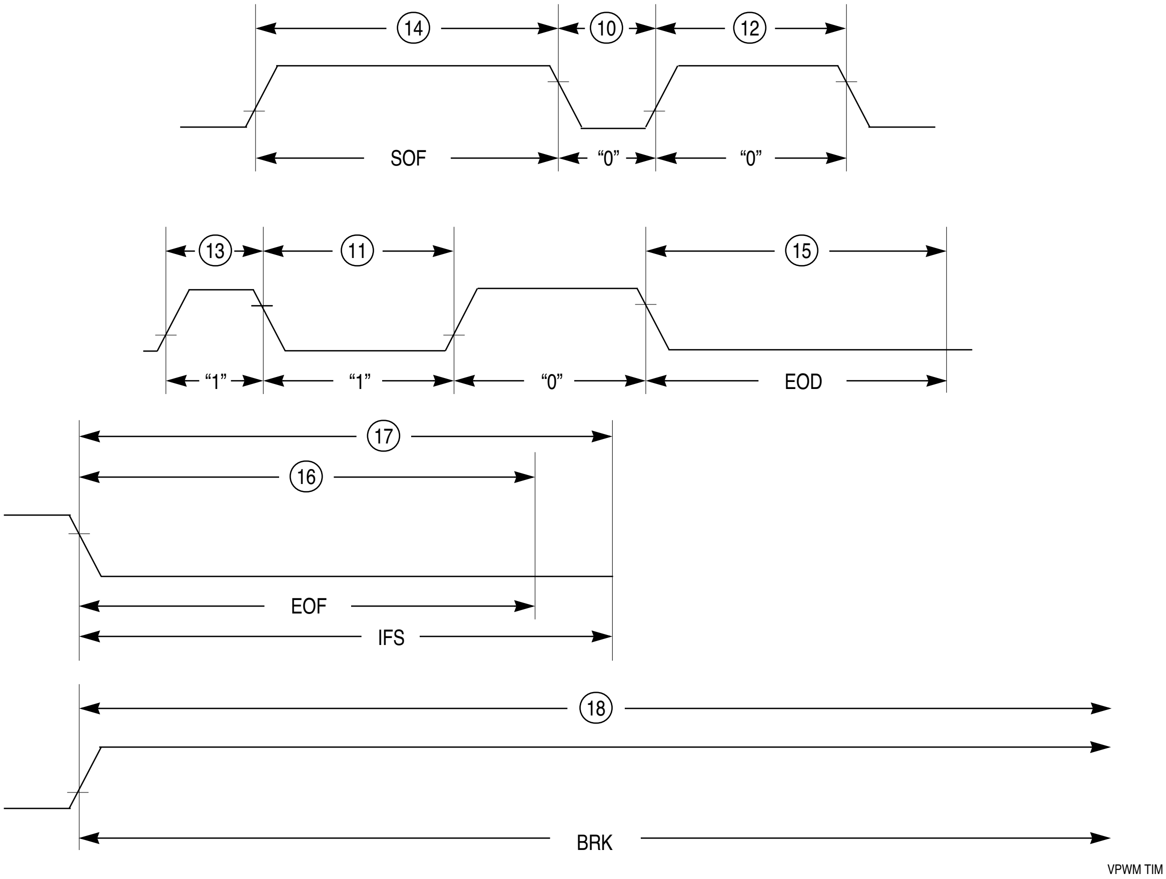


Figure A-6 Variable Pulse-Width Modulation (VPW) Symbol Timings

Table A-8 Transceiver Requirements (DC)

Symbol	Description	Conditions	Min	Max	Units
V_{OH}	Guaranteed output high voltage	100% bus utilization, 4 MHz (V_{BATT} = 9 to 26.5V)	6.25	8.0	V
		100% bus utilization, 4 MHz (V_{BATT} = 7 to 9V)	5.25	8.0	
$V_{OL, MAX}$	Maximum guaranteed output low voltage	100% bus utilization, 4 MHz	—	1.5	V
$V_{IL, MAX}$	Maximum input low voltage	V_{CC} = 4.75 to 5.25 V	—	3.5	V
$V_{IH, MIN}$	Minimum input high voltage	V_{CC} = 4.75 to 5.25 V	4.25	—	V
V_T	Nominal receiver trip point	–	3.875	—	V

Table A-9 Transmitter VPW Symbol Timings

(V_{BATT} = 12V, V_{CC} = 5.0V, V_{SS} = 0V, T_A = 25°C, unless otherwise noted.)

Characteristic	Number	Symbol	Min	Typical	Max	Unit
Passive logic 0	10	t_{TVP1}	58.0	64.0	70.0	µs
Passive logic 1	11	t_{TVP2}	122.0	128.0	134.0	µs
Active logic 0	12	t_{TVA1}	122.0	128.0	134.0	µs
Active logic 1	13	t_{TVA2}	58.0	64.0	70.0	µs
Start of frame (SOF)	14	t_{TVA3}	193.0	200.00	207.0	µs
End of data (EOD)	15	t_{TVP3}	193.0	200.0	207.0	µs
End of frame	16	t_{TV4}	271.0	280.0	289.0	µs
Inter-frame separator (IFS)	17	t_{TV6}	300.0	—	—	µs
Break (BRK)	18	t_{TV7}	—	1200	—	µs

Table A-10 Receiver VPW Symbol Timings

(V_{BATT} = 12V, V_{CC} = 5.0V, V_{SS} = 0V, T_A = 25°C, unless otherwise noted.)

Characteristic	Number	Symbol[1]	Min	Typical	Max	Unit
Passive logic 0	10	t_{RVP1}	34.0	64.0	96.0	µs
Passive logic 1	11	t_{RVP2}	96.0	128.0	163.0	µs
Active logic 0	12	t_{RVA1}	96.0	128.0	163.0	µs
Active logic 1	13	t_{RVA2}	34.0	64.0	96.0	µs
Start of frame (SOF)	14	t_{RVA3}	163.0	200.0	239.0	µs
End of data (EOD)	15	t_{RVP3}	163.0	200.0	239.0	µs
End of frame	16	t_{RV4}	239.0	280.0	320.0	µs
Break	18	t_{RV7}	768.0	–	–	µs

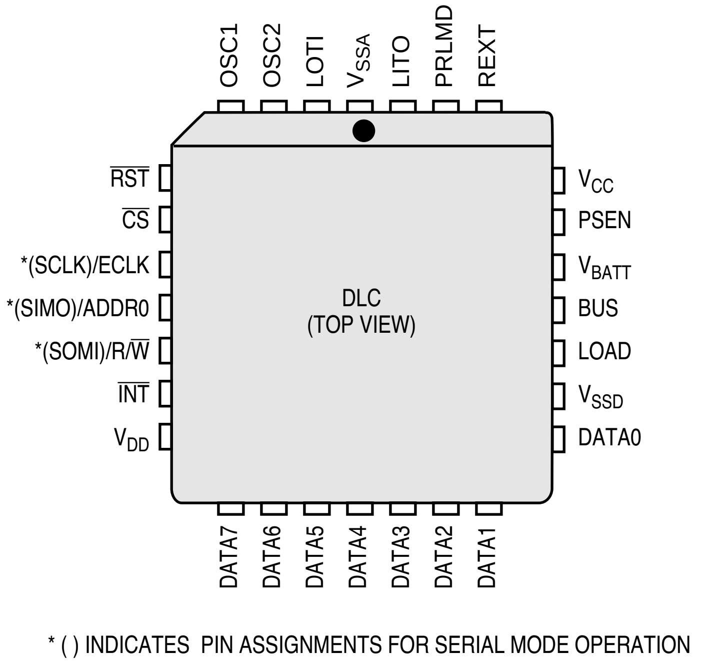
NOTES:

1. The receiver symbol timing boundaries are subject to an uncertainty of ± 1µs due to sampling considerations.

APPENDIX B MECHANICAL DATA AND ORDERING INFORMATION

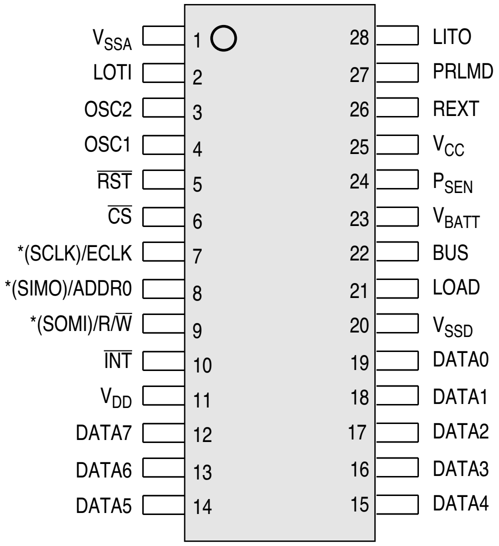
The MC68HC58 is available in two package options, a 28-pin PLCC (plastic leaded chip carrier) and a 28-pin SOIC (small outline integrated circuit). Refer to **Figures B-1** and **B-2**. **Figures B-3** and **B-4** show the corresponding dimensional drawings. Ordering information is available in **Table B-1**.

B.1 Pin Assignments



DLC PIN ASSIGNMENT

Figure B-1 MC68HC58 28-Pin PLCC

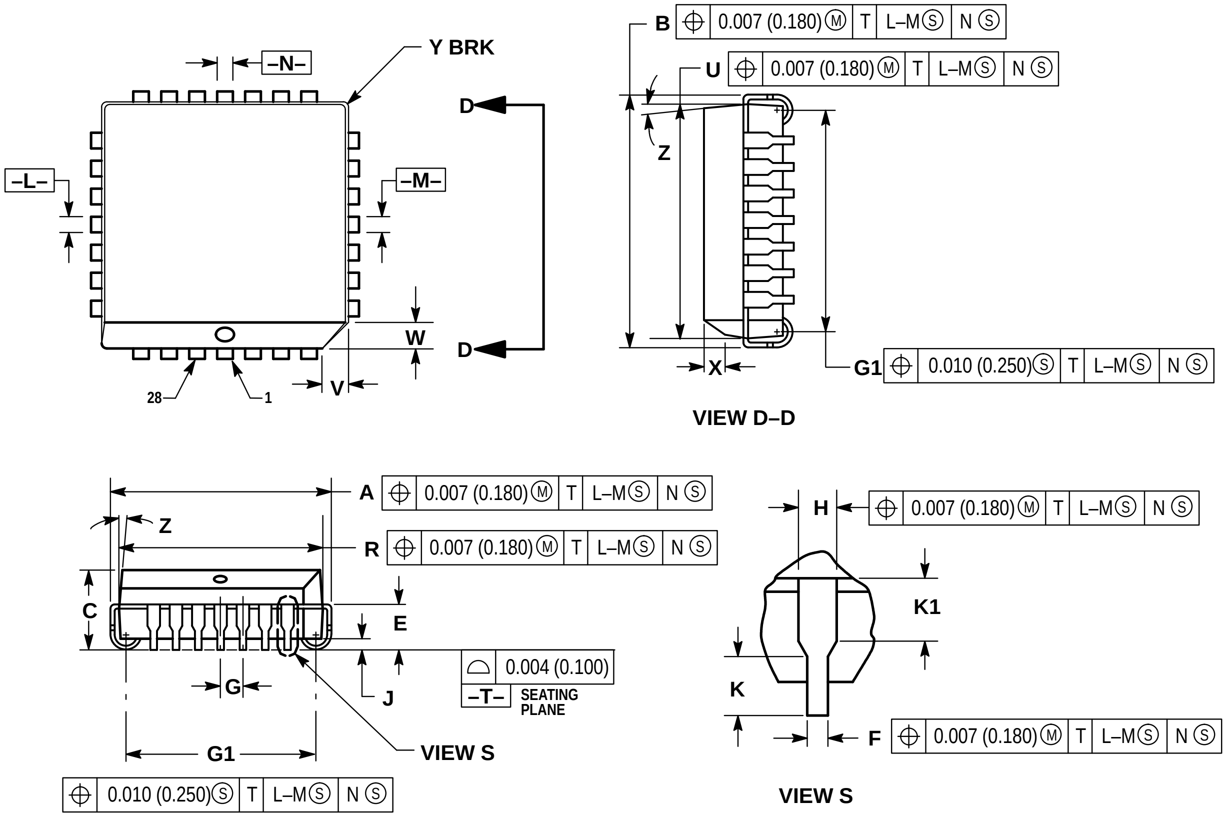


* ( ) INDICATES PIN ASSIGNMENTS FOR SERIAL MODE OPERATION

DLCP 28-PIN SOIC

Figure B-2 MC68HC58 28-Pin SOIC

5.5 Package Dimensions



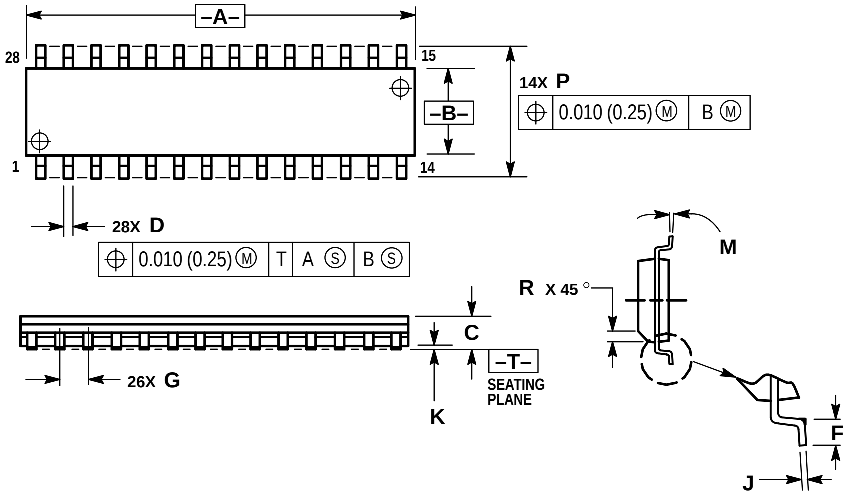
Y BRK
–N–
–L–
–M–
D
D
W
V
28
1
B 0.007 (0.180) Ⓜ T L–MⓈ N Ⓢ
U 0.007 (0.180) Ⓜ T L–MⓈ N Ⓢ
Z
X
G1 0.010 (0.250)Ⓢ T L–MⓈ N Ⓢ
VIEW D–D
A 0.007 (0.180) Ⓜ T L–MⓈ N Ⓢ
R 0.007 (0.180) Ⓜ T L–MⓈ N Ⓢ
Z
C
E
G
J
0.004 (0.100)
–T– SEATING PLANE
VIEW S
G1
0.010 (0.250)Ⓢ T L–MⓈ N Ⓢ
H 0.007 (0.180) Ⓜ T L–MⓈ N Ⓢ
K1
K
F 0.007 (0.180) Ⓜ T L–MⓈ N Ⓢ
VIEW S

NOTES:

1. DATUMS –L–, –M–, AND –N– DETERMINED WHERE TOP OF LEAD SHOULDER EXITS PLASTIC BODY AT MOLD PARTING LINE.
2. DIMENSION G1, TRUE POSITION TO BE MEASURED AT DATUM –T–, SEATING PLANE.
3. DIMENSIONS R AND U DO NOT INCLUDE MOLD FLASH. ALLOWABLE MOLD FLASH IS 0.010 (0.250) PER SIDE.
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
5. CONTROLLING DIMENSION: INCH.
6. THE PACKAGE TOP MAY BE SMALLER THAN THE PACKAGE BOTTOM BY UP TO 0.012 (0.300). DIMENSIONS R AND U ARE DETERMINED AT THE OUTERMOST EXTREMES OF THE PLASTIC BODY EXCLUSIVE OF MOLD FLASH, TIE BAR BURRS, GATE BURRS AND INTERLEAD FLASH, BUT INCLUDING ANY MISMATCH BETWEEN THE TOP AND BOTTOM OF THE PLASTIC BODY.
7. DIMENSION H DOES NOT INCLUDE DAMBAR PROTRUSION OR INTRUSION. THE DAMBAR PROTRUSION(S) SHALL NOT CAUSE THE H DIMENSION TO BE GREATER THAN 0.037 (0.940). THE DAMBAR INTRUSION(S) SHALL NOT CAUSE THE H DIMENSION TO BE SMALLER THAN 0.025 (0.635).

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.485	0.495	12.32	12.57
B	0.485	0.495	12.32	12.57
C	0.165	0.180	4.20	4.57
E	0.090	0.110	2.29	2.79
F	0.013	0.019	0.33	0.48
G	0.050 BSC		1.27 BSC	
H	0.026	0.032	0.66	0.81
J	0.020	—	0.51	—
K	0.025	—	0.64	—
R	0.450	0.456	11.43	11.58
U	0.450	0.456	11.43	11.58
V	0.042	0.048	1.07	1.21
W	0.042	0.048	1.07	1.21
X	0.042	0.056	1.07	1.42
Y	—	0.020	—	0.50
Z	2°	10°	2°	10°
G1	0.410	0.430	10.42	10.92
K1	0.040	—	1.02	—

Figure B-3 Case Outline #776-02



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.13 (0.005) TOTAL IN EXCESS OF D DIMENSION AT MAXIMUM MATERIAL CONDITION.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	17.80	18.05	0.701	0.711
B	7.40	7.60	0.292	0.299
C	2.35	2.65	0.093	0.104
D	0.35	0.49	0.014	0.019
F	0.41	0.90	0.016	0.035
G	1.27 BSC		0.050 BSC	
J	0.23	0.32	0.009	0.013
K	0.13	0.29	0.005	0.011
M	0 °	8 °	0 °	8 °
P	10.01	10.55	0.395	0.415
R	0.25	0.75	0.010	0.029

Figure B-4 Case Outline #751F-04

B.2 Obtaining Updated MC68HC58 Mechanical Information

Although all devices manufactured by Motorola conform to current JEDEC standards, complete mechanical information regarding MC68HC58 data link controller is available through Motorola’s Design-Net.

To download updated package specifications, perform the following steps:

1. Visit the Design-Net case outline database search engine at http://design-net.com/cgi-bin/cases.
2. Enter the case outline number, located in **Figures B-3** and **B-4** without the revision code (for example, 864A, not 864A-03) in the field next to the search button.
3. Download the file with the new package diagram.

B.3 Ordering Information

Table B-1 MC68HC58 Ordering Information

MC Order Information	Package	Description
MC68HC58	776-02	28-pin PLCC
	751F -04	28-pin SOIC

APPENDIX C DLC REGISTERS

C.1 Command Byte Register

CBR — Command Byte Register

	7	6	5	4	3	2	1	0
	GCOM			BTAD			RFC	
RESET:	0	0	0	0	0	0	0	0

Table C-1 General Command Summary (GCOM)

Value	Description
000	Do nothing
001	Enter standby mode
010	Send BREAK symbol
011	Send IFR on EOD with CRC
100	Terminate automatic retry
101	Send IFR on EOD without CRC
110	Reserved
111	Abort transmission

Table C-2 Byte Type and Destination Summary (BTAD)

Value	Description
000	Do not load
001	Load as transmit data
010	Reserved
011	Load as last byte of transmit data
100	Load as configuration byte
101	Load as first byte of transmit data
110	Load as configuration byte-immediate
111	Load as first and last byte of transmit data

Table C-3 RFC Field Encoding (RFC)

Value	Description
00	Do nothing
01	Reserved
10	Flush byte
11	Flush frame except for completion code

C.2 Configuration Byte Register

CBR — Configuration Byte Register

	7	6	5	4	3	2	1	0
	TM	TC		IMSK	IMOD	OSCD		4X
RESET:	0	0	0	0	0	1	1	0

Table C-4 Test Mode Control Bit (TM)

Value	Description
0	Normal operation
1	Test Mode

Table C-5 Test Configuration Field (TC)

Value	Description
00	Normal mode
–	Factory test

Table C-6 Interrupt Mask Bit (IMSK)

Value	Description
0	All interrupts to the MCU are enabled
1	All interrupts to the MCU are disabled

Table C-7 Interrupt Mode Bit (IMOD)

Value	Description
0	Default interrupts are the only ones enabled
1	Additional interrupt source added to default sources

Table C-8 Internal Clock Frequency Field (OSCD)

OSCD	Clock	External Clock			
Value	Divisor	2 MHz	4 MHz	6 MHz	8 MHz
00	1	2 MHz	4 MHz	6 MHz	8 MHz
01	2	1 MHz	2 MHz	3 MHz	4 MHz
10	3	0.66 MHz	1.33 MHz	2 MHz	2.66 MHz
11	4	500 kHz	1 MHz	1.5 MHz	2 MHz

Table C-9 High-Speed Control Bit (4X)

Value	Description
0	Normal clock division
1	4 times normal clock speed

C.3 Status Byte Register

SBR — Status Byte Register

	7	6	5	4	3	2	1	0
	RFS			DLI	NETF	4XMD	TMFS	
RESET:	0	0	0	X	0	0	0	0

Table C-10 Receive FIFO Status Field Encoding (RFS)

Value	Description
000	Buffer invalid or empty
001	Buffer contains more than one byte
010	Buffer contains a completion code
011	Data byte in 13th buffer position, no completion code
100	One data byte in buffer
101	Completion code at head of buffer, more bytes available
110	Completion code at head of buffer, another frame available
111	Completion code only at head of buffer

Table C-11 Data Link Idle Bit (DLI)

Value	Description
0	SAE J1850 bus is active
1	SAE J1850 bus is idle

Table C-12 Network Fault Bit (NETF)

Value	Description
0	Active state occurred before 60 µs
1	Active state not sensed after 60 µs

Table C-13 4X Mode Bit (4XMD)

Value	Description
0	DLCP/S operating in normal mode
1	DLCP/S operating at 41.67 kbps

Table C-14 Transmit FIFO Status Field Encoding (TMFS)

Value	Description
00	Buffer empty
01	Buffer contains data bytes
10	Buffer almost full
11	Buffer full

C.4 Completion Code Byte Register

CCBR — Completion Code Byte Register

	7	6	5	4	3	2	1	0
	ERRF	RFO	TMS		IFR	IFRC	ERRC	
RESET	0	0	0	0	0	0	0	0

Table C-15 Error Bit (ERRF)

Value	Description
0	No error occurred
1	Error occurred

Table C-16 Receive FIFO Overrun Bit (RFO)

Value	Description
0	No receiver buffer overrun occurred
1	Receiver buffer overrun occurred

Table C-17 Transmitter Status Field Encoding (TMS)

Value	Description
00	Transmitter not involved
01	Transmitter underrun
10	Transmitter lost arbitration
11	Transmitter successful

Table C-18 In-Frame Response Bit (IFR)

Value	Description
0	Not an IFR
1	IFR

Table C-19 In-Frame Response CRC Bit (IFRC)

Value	Description
0	IFR without CRC
1	IFR with CRC

Table C-20 Error Code Field Encoding

Value	Description
00	CRC error
01	Incomplete byte
10	Bit timing error
11	Break

INDEX