1

AS 15531/MIL-STD-1553B Digital Time Division Command/Response Multiplex Data Bus

Chris deLong
Honeywell, Defense Avionics Systems

1.1 Introduction

MIL-STD-1553 is a standard which defines the electrical and protocol characteristics for a data bus. SAE AS-15531 is the commercial equivalent to the military standard. A data bus is similar to what the personal computer and office automation industry have dubbed a "Local Area Network (LAN)." In avionics, a data bus is used to provide a medium for the exchange of data and information between various systems and subsystems.

1.1.1 Background

In the 1950s and 1960s, avionics were simple standalone systems. Navigation, communications, flight controls, and displays consisted of analog systems. Often, these systems were composed of multiple boxes interconnected to form a single system. The interconnections between the various boxes was accomplished with point-to-point wiring. The signals mainly consisted of analog voltages, synchro-resolver signals, and relay/switch contacts. The location of these boxes within the aircraft was a function of operator need, available space, and the aircraft weight and balance constraints. As more and more systems were added,

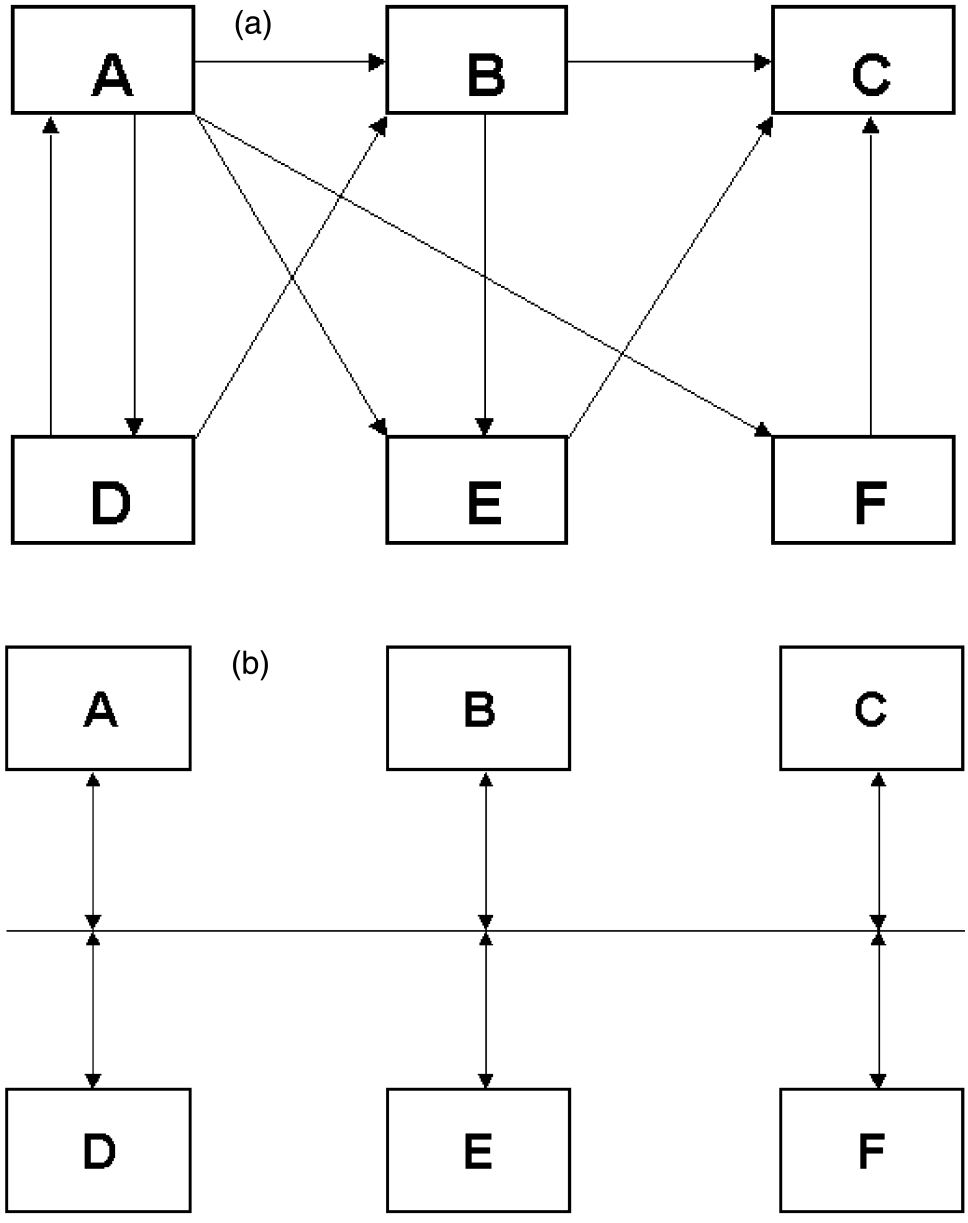

FIGURE 1.1 Systems configurations.

the cockpits became crowded due to the number of controls and displays, and the overall weight of the aircraft increased.

By the late 1960s and early 1970s, it was necessary to share information between various systems to reduce the number of black boxes required by each system. A single sensor providing heading and rate information could provide those data to the navigation system, the weapons system, the flight control system, and pilot's display system (see Figure 1.1a). However, the avionics technology was still basically analog, and while sharing sensors did produce a reduction in the overall number of black boxes, the interconnecting signals became a "rat's nest" of wires and connectors. Moreover, functions or systems

that were added later became an integration nightmare as additional connections of a particular signal could have potential system impacts, plus since the system used point-to-point wiring, the system that was the source of the signal typically had to be modified to provide the additional hardware needed to output to the newly added subsystem. As such, intersystem connections were kept to the bare minimums.

By the late 1970s, with the advent of digital technology, digital computers had made their way into avionics systems and subsystems. They offered increased computational capability and easy growth, compared to their analog predecessors. However, the data signals — the inputs and outputs from the sending and receiving systems — were still mainly analog in nature, which led to the configuration of a small number of centralized computers being interfaced to the other systems and subsystems via complex and expensive analog-to-digital and digital-to-analog converters.

As time and technology progressed, the avionics systems became more digital. And with the advent of the microprocessor, things really took off. A benefit of this digital application was the reduction in the number of analog signals, and hence the need for their conversion. Greater sharing of information could be provided by transferring data between users in digital form. An additional side benefit was that digital data could be transferred bidirectionally, whereas analog data were transferred unidirectionally. Serial rather than parallel transmission of the data was used to reduce the number of interconnections within the aircraft and the receiver/driver circuitry required with the black boxes. But this alone was not enough. A data transmission medium which would allow all systems and subsystems to share a single and common set of wires was needed (see Figure 1.1b). By sharing the use of this interconnect, the various subsystems could send data between themselves and to other systems and subsystems, one at a time, and in a defined sequence. Enter the 1553 Data Bus.

1.1.2 History and Applications

MIL-STD-1553(USAF) was released in August of 1973. The first user of the standard was the F-16. Further changes and improvements were made and a tri-service version, MIL-STD-1553A, was released in 1975. The first user of the "A" version of the standard was again the Air Force's F-16 and the Army's new attack helicopter, the AH-64A Apache. With some "real world" experience, it was soon realized that further definitions and additional capabilities were needed. The latest version of the standard, 1553B, was released in 1978.

Today the 1553 standard is still at the "B" level; however, changes have been made. In 1980, the Air Force introduced Notice 1. Intended only for Air Force applications, Notice 1 restricted the use of many of the options within the standard. While the Air Force felt this was needed to obtain a common set of avionics systems, many in industry felt that Notice 1 was too restrictive and limited the capabilities in the application of the standard. Released in 1986, the tri-service Notice 2 (which supersedes Notice 1) places tighter definitions upon the options within the standard. And while not restricting an option's use, it tightly defines how an option will be used if implemented. Notice 2, in an effort to obtain a common set of operational characteristics, also places a minimum set of requirements upon the design of the black box. The military standard was converted to its commercial equivalent as SAE AS 15531, as part of the government's effort to increase the use of commercial products.

Since its inception, MIL-STD-1553 has found numerous applications. Notice 2 even removed all references to "aircraft" or "airborne" so as not to limit its applications. The standard has also been accepted and implemented by NATO and many foreign governments. The U.K. has issued Def Stan 00-18 (Part 2) and NATO has published STANAG 3838 AVS, both of which are versions of MIL-STD-1553B.

1.2 The Standard

MIL-STD-1553B defines the term Time Division Multiplexing (TDM) as "the transmission of information from several signal sources through one communications system with different signal samples staggered in time to form a composite pulse train." For our example in Figure 1.1b, this means that data can be transferred between multiple avionics units over a single transmission media, with the communications

between the different avionics boxes taking place at different moments in time, hence time division. Table 1.1 is a summary of the 1553 Data Bus Characteristics. However, before defining how the data are transferred, it is necessary to understand the data bus hardware.

1.2.1 Hardware Elements

The 1553 standard defines certain aspects regarding the design of the data bus system and the black boxes to which the data bus is connected. The standard defines four hardware elements: transmission media, remote terminals, bus controllers, and bus monitors; each of which is detailed as follows.

TABLE 1.1 Summary of the 1553 Data Bus Characteristics

Data Rate	1 MHz
Word Length	20 bits
Data Bits per Word	16 bits
Message Length	Maximum of 32 data words
Transmission Technique	Half-Duplex
Operation	Asynchronous
Encoding	Manchester II Bi-phase
Protocol	Command-Response
Bus Control	Single or Multiple
Message Formats	Controller-to-Terminal (BC-RT)
	Terminal-to-Controller (RT-BC)
	Terminal-to-Terminal (RT-RT)
	Broadcast
	System Control
Number of Remote Terminals	Maximum of 31
Terminal Types	Remote Terminal (RT)
	Bus Controller (BC)
	Bus Monitor (BM)
Transmission Media	Twisted Shielded Pair Cable
Coupling	Transformer or Direct

1.2.1.1 Transmission Media

The transmission media, or data bus, is defined as a twisted shielded pair transmission line consisting of the main bus and a number of stubs. There is one stub for each terminal (system) connected to the bus. The main data bus is terminated at each end with a resistance equal to the cable's characteristic impedance. This termination makes the data bus behave electrically like an infinite transmission line. Stubs, which are added to the main bus in order to connect the terminals, provide "local" loads, and produce an impedance mismatch where added. This mismatch, if not properly controlled, produces electrical reflections and degrades the performance of the main bus. Therefore, the characteristics of both the main bus and the stubs are specified within the standard. Table 1.2 is a summary of the transmission media characteristics.

The standard specifies two stub methods: direct and transformer coupled. This refers to the method in which a terminal is connected to the main bus. Figure 1.2 shows the two methods, the primary difference between the two being that the transformer coupled method utilizes an isolation transformer for connecting the stub cable to the main bus cable. In both methods, two isolation resistors are placed in series with the bus. In the direct coupled method, the resistors are typically located within the terminal, whereas in the transformer coupled method, the resistors are typically located with the coupling transformer in boxes called data bus couplers. A variety of couplers are available, providing single or multiple stub connections.

Another difference between the two coupling methods is the length of the stub. For the direct coupled method, the stub length is limited to a maximum of 1 ft. For the transformer coupled method, the stub can be up to a maximum length of 20 ft. Therefore for direct coupled systems, the data bus must be

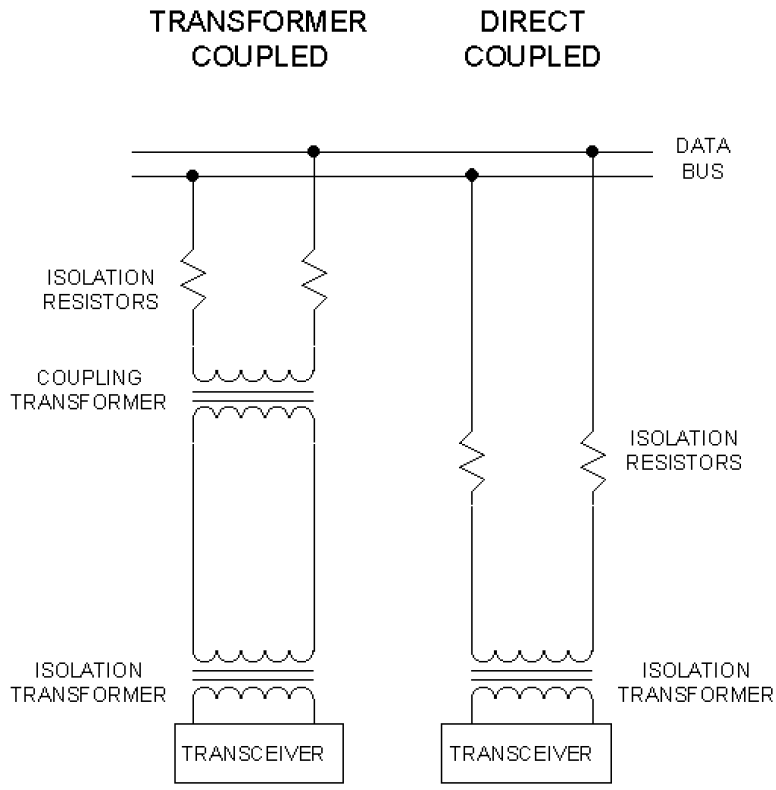

FIGURE 1.2 Terminal connection methods.

routed in close proximity to each of the terminals, whereas for a transformer coupled system, the data bus may be up to 20 ft away from each terminal.

TABLE 1.2 Summary of Transmission Media Characteristics

Cable Type	Twisted Shielded Pair
Capacitance	30.0 pF/ft max — wire to wire
Characteristic Impedance	70.0 to 85.0 ohms at 1 MHz
Cable Attenuation	1.5 dbm/100 ft at 1 MHz
Cable Twists	4 twists per foot maximum
Shield Coverage	90% minimum
Cable Termination	Cable impedance (±2%)
Direct Coupled Stub Length	Maximum of 1 ft
Transformer Coupled Stub Length	Maximum of 20 ft

1.2.1.2 Remote Terminal

A remote terminal is defined within the standard as "All terminals not operating as the bus controller or as a bus monitor." Therefore if it is not a controller, monitor, or the main bus or stub, it must be a remote terminal — sort of a "catch all" clause. Basically, the remote terminal is the electronics necessary to transfer data between the data bus and the subsystem. So what is a subsystem? For 1553 applications, the subsystem is the sender or user of the data being transferred.

In the earlier days of 1553, remote terminals were used mainly to convert analog and discrete data to/from a data format compatible with the data bus. The subsystems were still the sensor which provided the data and computer which used the data. As more and more digital avionics became available, the

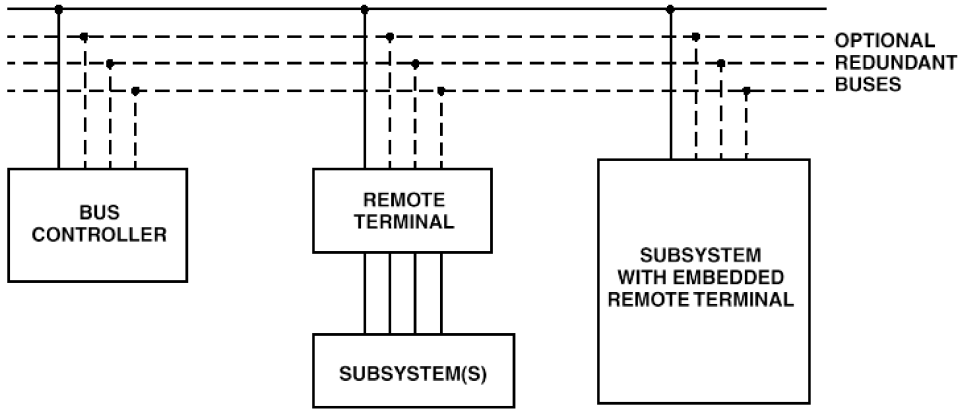

FIGURE 1.3 Simple multiplex architecture.

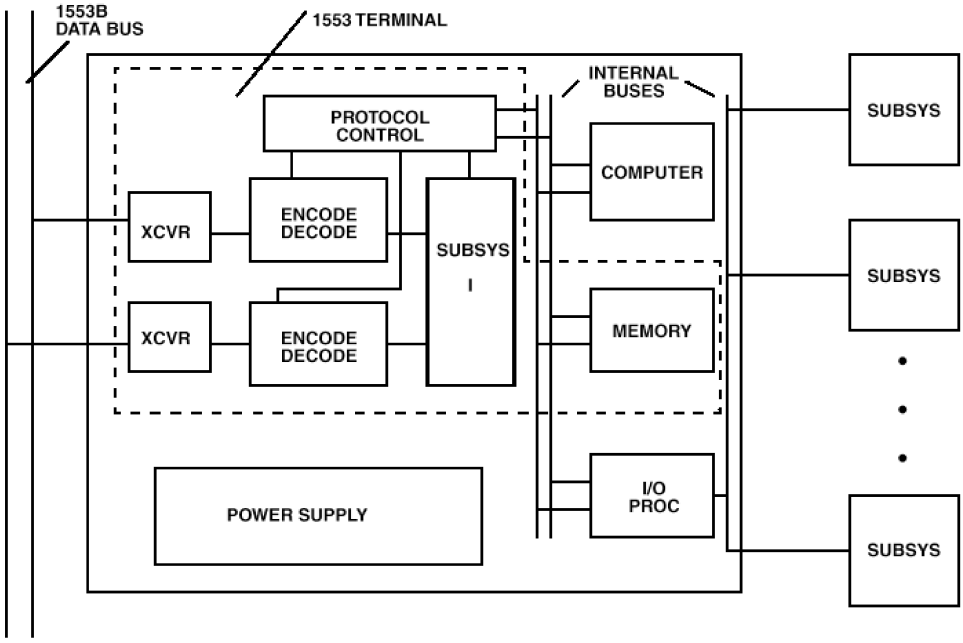

FIGURE 1.4 Terminal definition.

trend has been to embed the remote terminal into the sensor and computer. Today it is common for the subsystem to contain an embedded remote terminal. Figure 1.3 shows the different levels of remote terminals possible.

A remote terminal typically consists of a transceiver, an encoder/decoder, a protocol controller, a buffer or memory, and a subsystem interface. In a modern black box containing a computer or processor, the subsystem interface may consist of the buffers and logic necessary to interface to the computer's address, data, and control buses. For dual redundant systems two transceivers and two encoders/decoders would be required to meet the requirements of the standard.

Figure 1.4 is a block diagram of a remote terminal and its connection to a subsystem. In short, the remote terminal consists of all the electronics necessary to transfer data between the data bus and the user or originator of the data being transferred.

But a remote terminal is more than just a data formatter. It must be capable of receiving and decoding commands from the bus controller, and respond accordingly. It must also be capable of buffering a message-worth of data, be capable of detecting transmission errors and performing validation tests upon the data, and reporting the status of the message transfer. A remote terminal must be capable of performing a few of the bus management commands (referred to as mode commands), and for dual redundant applications it must be capable of listening to and decoding commands on both buses at the same time.

A remote terminal must strictly follow the protocol as defined by the standard. It can only respond to commands received from the bus controller (i.e., it only speaks when spoken to). When it receives a valid command, it must respond within a defined amount of time. If a message does not meet the validity requirements defined, then the remote terminal must invalidate the message and discard the data (not allow the data to be used by the subsystem). In addition to reporting status to the bus controller, most remote terminals today are also capable of providing some level of status information to the subsystem regarding the data received.

1.2.1.3 Bus Controller

The bus controller is responsible for directing the flow of data on the bus. While several terminals may be capable of performing as the bus controller, only one bus controller is allowed to be active at any one time. The bus controller is the only device allowed to issue commands onto the data bus. The commands may be for the transfer of data, or the control and management of the bus (referred to as mode commands).

Typically, the bus controller is a function that is contained within some other computer, such as a mission computer, a display processor, or a fire control computer. The complexity of the electronics associated with the bus controller is a function of the subsystem interface (the interface to the computer), the amount of error management and processing to be performed, and the architecture of the bus controller. There are three types of bus controllers architectures: a word controller, a message controller, and a frame controller.

A *word controller* is the oldest and simplest type. Few word controllers are built today and they are only mentioned herein for completeness. For a word controller, the terminal electronics transfers one word at a time to the subsystem. Message buffering and validation must be performed by the subsystem.

Message controllers output a single message at a time, interfacing with the computer only at the end of the message or perhaps when an error occurrs. Some message controllers are capable of performing minor error processing, such as transmitting once on the alternate data bus, before interrupting the computer. The computer will inform the interface electronics of where the message exists in memory and provide a control word. For each message the control word typically informs the electronics of the message type (e.g., an RT-BC or RT-RT command), which bus to use to transfer the message, where to read or write the data words in memory, and what to do if an error occurs. The control words are a function of the hardware design of the electronics and are not standardized among bus controllers.

A *frame controller* is the latest concept in bus controllers. A frame controller is capable of processing multiple messages in a sequence defined by the computer. The frame controller is typically capable of error processing as defined by the message control word. Frame controllers are used to "off-load" the computer as much as possible, interrupting only at the end of a series of messages or when an error it can not handle is detected.

There is no requirement within the standard as to the internal workings of a bus controller, only that it issue commands onto the bus.

1.2.1.4 Bus Monitor

A bus monitor is just that. A terminal which listens to (monitors) the exchange of information on the data bus. The standard strictly defines what bus monitors may be used for, stating that the information obtained by a bus monitor be used "for off-line applications (e.g., flight test recording, maintenance recording or mission analysis) or to provide a back-up bus controller sufficient information to take over as the bus controller." Monitors may collect all the data from the bus or may collect selected data.

The reason for restricting its use is that while a monitor may collect data, it deviates from the command-response protocol of the standard in that a monitor is a passive device that does not transmit a status word, and therefore can not report on the status of the information transferred. Therefore, bus monitors fall into two categories: a recorder for testing, or as a terminal functioning as a back-up bus controller.

In collecting data, a monitor must perform the same message validation functions as the remote terminal and, if an error is detected, inform the subsystem of the error (the subsystem may still record the data, but the error should be noted). For monitors which function as recorders for testing, the subsystem is typically a recording device or a telemetry transmitter. For monitors which function as back-up bus controllers, the subsystem is the computer.

Today it is common that bus monitors also contain a remote terminal. When the monitor receives a command addressed to its terminal address, it responds as a remote terminal. For all other commands, it functions as a monitor. The remote terminal portion could be used to provide feedback to the bus controller of the monitor's status, such as the amount of memory or time left, or to reprogram a selective monitor as to what messages to capture.

1.2.1.5 Terminal Hardware

The electronic hardware between a remote terminal, bus controller, and bus monitor does not differ much. Both the remote terminal and bus controller (and bus monitor if it is also a remote terminal) must have the transmitters/receivers and encoders/decoders to format and transfer data. The requirements upon the transceivers and the encoders/decoders do not vary between the hardware elements. Table 1.3 lists the electrical characteristics of the terminals.

All three elements have some level of subsystem interface and data buffering. The primary difference lies in the protocol control logic and often this just a different series of micro-coded instructions. For this reason, it is common to find 1553 hardware circuitry that is also capable of functioning as all three devices.

TABLE 1.3 Terminal Electrical Characteristics

Requirement	Transformer Coupled	Direct Coupled	Condition
	Input Characteristics		
Input Level	0.86–14.0 V	1.2–20.0 V	p–p, l–l
No Response	0.0–0.2 V	0.0–0.28 V	p–p, l–l
Zero Crossing Stability	±150.0 nsec	±150.0 nsec	
Rise/Fall Times	0 nsec	0 nsec	Sine Wave
Noise Rejection	140.0 mV WGN[a]	200.0 mV WGN	BER 1[b] per 10^7
Common Mode Rejection	±10.0 V peak	±10.0 V peak	line-gnd, DC-2.0 MHz
Input Impedance	1000 ohms	2000 ohms	75 kHz–1 MHz
	Output Characteristics		
Output Level	18.0–27.0 V	6.0–9.0 V	p–p, l–l
Zero Crossing Stability	25.0 nsec	25.0 nsec	
Rise/Fall Times	100–300 nsec	100–300 nsec	10%–90%
Maximum Distortion	±900.0 mV	±300.0 mV	peak, l–l
Maximum Output Noise	14.0 mV	5.0 mV	rms, l–l
Maximum Residual Voltage	±250.0 mV	±90.0 mV	peak, l–l

[a] WGN = White Gaussian Noise.
[b] BER = Bit Error Rate.

There is an abundance of "off-the-shelf" components available today from which to design a terminal. These vary from discrete transceivers, encoders/decoders, and protocol logic devices to a single dual redundant hybrid containing everything but the transformers.

1.3 Protocol

The rules under which the transfers occur is referred to as “protocol”. The control, data flow, status reporting, and management of the bus is provided by three word types.

1.3.1 Word Types

Three distinct word types are defined by the standard. These are command words, data words, and status words. Each word type has a unique format yet all three maintain a common structure. Each word is 20 bits in length. The first three bits are used as a synchronization field, thereby allowing the decode clock to re-sync at the beginning of each new word. The following 16 bits are the information field and differ among the three word types. The last bit is the parity bit. Parity is based on odd parity for the single word. The three word types are shown in Figure 1.5.

Bit encoding for all words is based on bi-phase Manchester II format. The Manchester II format provides a self-clocking waveform in which the bit sequence is independent. The positive and negative voltage levels of the Manchester waveform is DC balanced (same amount of positive signal as there is negative signal) and as such is well suited for transformer coupling. A transition of the signal occurs at the center of the bit time. A logic “0” is a signal that transitions from a negative level to a positive level. A logic “1” is a signal that transitions from a positive level to a negative level.

The terminal’s hardware is responsible for the Manchester encoding and decoding of the word types. The interface that the subsystem sees is the 16-bit information field of all words. The sync and parity fields are not provided directly. However, for received messages, the decoder hardware provides a signal to the protocol logic as to the sync type the word was and as to whether parity was valid or not. For transmitted messages, there is an input to the encoder as to what sync type to place at the beginning of the word, and parity is automatically calculated by the encoder.

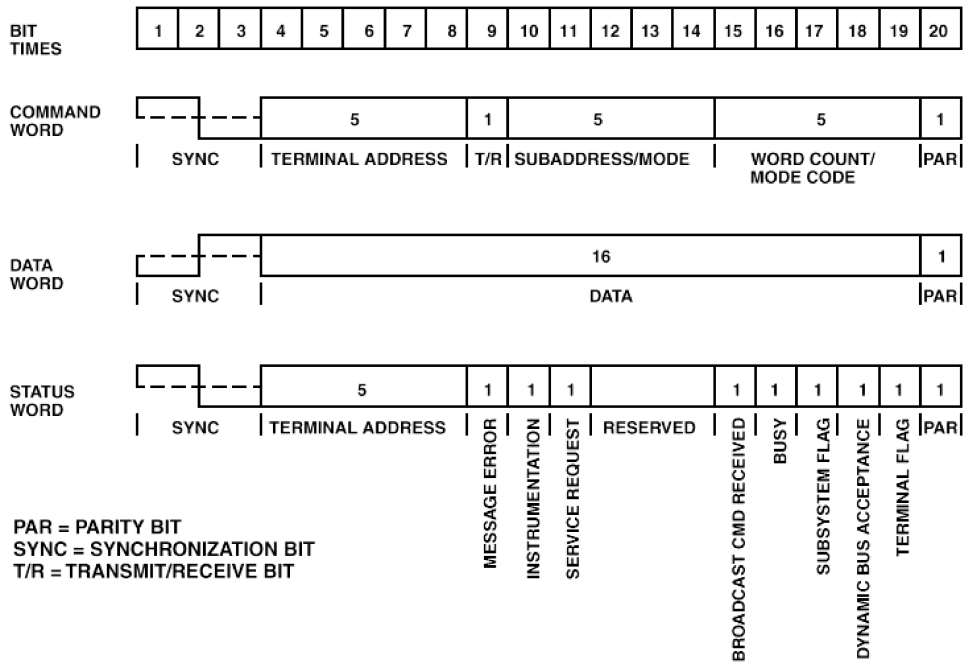

FIGURE 1.5 Word formats.

1.3.1.1 Sync Fields

The first three bit times of all word types is called the sync field. The sync waveform is in itself an invalid Manchester waveform as the transition only occurs at the middle of the second bit time. The use of this distinct pattern allows the decoder to re-sync at the beginning of each word received and maintain the overall stability of the transmissions.

Two distinct sync patterns are used: the command/status sync, and the data sync. The command/status sync has a positive voltage level for the first one and a half bit times, then transitions to a negative voltage level for the second one and a half bit times. The data sync is the opposite — a negative voltage level for the first one and a half bit times, then transitions to a positive voltage level for the second one and a half bit times. The sync patterns are shown in Figure 1.5.

1.3.1.2 Command Word

The Command Word (CW) specifies the function that a remote terminal(s) is to perform. This word is only transmitted by the active bus controller. The word begins with a command sync in the first three bit times. The following 16-bit information field is as defined in Figure 1.5.

The five-bit Terminal Address (TA) field (bit times 4–8) states to which unique remote terminal the command is intended (no two terminals may have the same address). Note that an address of 00000 is a valid address, and that an address of 11111 is reserved for use as the broadcast address. Also note that there is no requirement that the bus controller be assigned an address, therefore the maximum number of terminals the data bus can support is 31. Notice 2 to the standard requires that the terminal address be wire programmable externally to the black box (i.e., an external connector) and that the remote terminal electronics perform a parity test upon the wired terminal address. The Notice basically states that an open circuit on an address line is detected as a logic "1," that connecting an address line to ground is detected as a logic "0," and that odd parity will be used in testing the parity of the wired address field.

The next bit (bit time 9) is the Transmit/Receive (T/R) bit. This defines the direction of information flow and is always from the point of view of the remote terminal. A transmit command (logic 1) indicates that the remote terminal is to transmit data, while a receive command (logic 0) indicates that the remote terminal is going to receive data. The only exceptions to this rule are associated with mode commands.

The following five bits (bit times 10–14) are the Subaddress (SA)/Mode Command (MC) bits. Logic 00000 or 11111 within this field shall be decoded to indicate that the command is a Mode Code Command. All other logic combinations of this field are used to direct the data to different functions within the subsystem. An example might be that 00001 is position and rate data, 00010 is frequency data, 10010 is display information, and 10011 is self-test data. The use of the subaddresses is left to the designer, however, Notice 2 suggests the use of subaddress 30 for data wraparound.

The next five bit positions (bit times 15–19) define the Word Count (WC) or Mode Code to be performed. If the Subaddress/Mode Code field was 00000 or 11111, then this field defines the mode code to be performed. If not a mode code, then this field defines the number of data words either to be received or transmitted depending on the T/R bit. A word count field of 00000 is decoded as 32 data words.

The last bit (bit time 20) is the word parity bit. Only odd parity shall be used.

1.3.1.3 Data Word

The Data Word (DW) contains the actual information that is being transferred within a message. Data words can be transmitted by either a remote terminal (transmit command) or a bus controller (receive command). The first three bit times contain a data sync. This sync pattern is the opposite of that used for command and status words and therefore is unique to the data word type.

The following 16 bits of information are left to the designer to define. The only standard requirement is that the most significant bit (MSB) of the data be transmitted first. While the standard provides no guidance as to their use, Section 80 of MIL-HDBK-1553A and SAE AS-15532 provides guidance and lists the formats (i.e., bit patterns, resolutions, etc.) of the most commonly used data words.

The last bit (bit time 20), is the word parity bit. Only odd parity shall be used.

1.3.1.4 Status Word

The Status Word (SW) is only transmitted by a remote terminal in response to a valid message. The status word is used to convey to the bus controller whether a message was properly received or the state of the remote terminal (i.e., service request, busy, etc.). The status word is defined in Figure 1.5. Since the status word conveys information to the bus controller, there are two views as to the meaning of each bit — what the setting of the bit means to a remote terminal, and what the setting of the bit means to a bus controller. Each field of the status word, and its potential meanings, is examined below.

1.3.1.4.1 Resetting the Status Word

The Status Word, with the exception of the remote terminal address, is cleared after receipt of a valid command word. The two exceptions to this rule are if the command word received is a Transmit Status Word Mode Code or a Transmit Last Command Word Mode Code. Conditions which set the individual bits of the word may occur at any time. If after clearing the status word, the conditions for setting the bits still exists, then the bits shall be set again.

Upon detection of a error in the data being received, the Message Error bit is set and the transmission of the status word is suppressed. The transmission of the status word is also suppressed upon receipt of a broadcast message. For an illegal message (i.e., an illegal Command Word), the Message Error bit is set and the status word is transmitted.

1.3.1.4.2 Status Word Bits

Terminal Address. The first five bits (bit times 4–8) of the information field are the Terminal Address (TA). These five bits should match the corresponding field within the command word that the terminal received. The remote terminal sets these bit to the address to which it has been programmed. The bus controller should examine these bits to insure that the terminal responding with its status word was indeed the terminal to which the command word was addressed. In the case of a remote terminal to remote terminal message (RT-RT), the receiving terminal should compare the address of the second command word with that of the received status word. While not required by the standard, it is good design practice to insure that the data received are from a valid source.

Message Error. The next bit (bit time 9) is the Message Error (ME) bit. This bit is set to a logic "1" by the remote terminal upon detection of a error in the message or upon detection of an invalid message (i.e., Illegal Command) to the terminal. The error may occur in any of the data words within the message. When the terminal detects an error and sets this bit, none of the data received within the message shall be used. If an error is detected within a message and the ME bit is set, the remote terminal must suppress the transmission of the status word (see Resetting of the Status Word). If the terminal detected an illegal command, the ME bit is set and the status word is transmitted. All remote terminals must implement the ME bit in the status word.

Instrumentation. The Instrumentation bit (bit time 10) is provided so as to differentiate between a command word and a status word (remember, they both have the same sync pattern). The instrumentation bit in the status word is always set to logic "0." If used, the corresponding bit in the command word is set to a logic "1." This bit in the command word is the most significant bit of the subaddress field, and therefore would limit the subaddresses used to 10000–11110, hence reducing the number of subaddresses available from 30 to 15. The instrumentation bit is also the reason why there are two mode code indentifiers (00000 and 11111), the latter required when the instrumentation bit is used.

Service Request. The Service Request bit (bit time 11) is such that the remote terminal can inform the bus controller that it needs to be serviced. This bit is set to a logic "1" by the subsystem to indicate that servicing is needed. This bit is typically used when the bus controller is "polling" terminals to determine if they require processing. The bus controller upon receiving this bit set to a logic "1" typically does one of the following. It can take a predetermined action such as issuing a series of messages, or it can request further data from the remote terminal as to its needs. The later can be accomplished by requesting the terminal to transmit data from a defined subaddress or by using the Transit Vector Word Mode Code.

Reserved. Bit times 12–14 are reserved for future growth of the standard and must be set to a logic "0." The bus controller should declare a message in error if the remote terminal responds with any of these bits set in its status word.

Broadcast Command Received. The Broadcast Command Received bit (bit time 15) indicates that the remote terminal received a valid broadcast command. Upon receipt of a valid broadcast command, the remote terminal sets this bit to logic "1" and suppresses the transmission of its status words. The bus controller may issue a Transmit Status Word or Transmit Last Command Word Mode Code to determine if the terminal received the message properly.

Busy. The Busy bit (bit time 16) is provided as a feedback to the bus controller as to when the remote terminal is unable to move data between the remote terminal electronics and the subsystem in compliance to a command from the bus controller.

In the earlier days of 1553, the Busy bit was required because many of the subsystem interfaces (analogs, synchros, etc.) were much slower compared to the speed of the multiplex data bus. Some terminals were not able to move the data fast enough. So instead of potentially losing data, a terminal was able to set the Busy bit, indicating to the bus controller is could not handle new data at that time, and for the bus controller to try again later. As new systems have been developed, the need for the use of Busy has been reduced. However, there are systems that still need and have a valid use for the Busy bit. Examples of these are radios, where the bus controller issues a command to the radio to tune to a certain frequency. It may take the radio several seconds to accomplish this, and while it is tuning it may set the Busy bit to inform the bus controller that it is doing as it was told.

When a terminal is busy, it does not need to respond to commands in the "normal" way. For receive commands the terminal collects the data, but does not have to pass the data to the subsystem. For transmit commands, the terminal transmits its status word only. Therefore, while a terminal is busy the data it supplies to the rest of the system are not available. This can have an overall effect upon the flow of data within the system and may increase the data latency within time-critical systems (e.g., flight controls).

Some terminals used the Busy bit to overcome design problems, setting the Busy bit whenever needed. Notice 2 to the standard "strongly discourages" the use of the Busy bit. However, as shown in the example above, there are valid needs for its use. Therefore, if used, Notice 2 now requires that the Busy bit may only be set as the result of a particular command received from the bus controller and not due to an internal periodic or processing function. By following this requirement, the bus controller, with prior knowledge of the remote terminal's characteristics, can determine what will cause a terminal to go busy and minimize the effects on data latency throughout the system.

Subsystem Flag. The Subsystem Flag bit (bit time 17) is used to provide "health" data regarding the subsystems to which the remote terminal is connected. Multiple subsystems may logically "OR" their bits together to form a composite health indicator. This single bit is only to serve as an indicator to the bus controller and user of the data that a fault or failure exists. Further information regarding the nature of the failure must be obtained in some other fashion. Typically, a subaddress is reserved for built-in-test (BIT) information, with one or two words devoted to subsystem status data.

Dynamic Bus Control Acceptance. The Dynamic Bus Control Acceptance bit (bit time 18) is used to inform the bus controller that the remote terminal has received the Dynamic Bus Control Mode Code and has accepted control of the bus. For the remote terminal, the setting of this bit is controlled by the subsystem and is based upon passing some level of built-in-test (i.e., a processor passing its power-up and continuous background tests).

The remote terminal upon transmitting its status word becomes the bus controller. The bus controller, upon receipt of the status word from the remote terminal with this bit set, ceases to function as the bus controller and may become a remote terminal or bus monitor.

Terminal Flag. The Terminal Flag bit (bit time 19) is used to inform the bus controller of a fault or failure within the remote terminal circuitry (only the remote terminal). A logic "1" shall indicate a fault condition. This bit is used solely to inform the bus controller of a fault or failure. Further information

regarding the nature of the failure must be obtained in some other fashion. Typically, a subaddress is reserved for BIT information, or the bus controller may issue a Transmit BIT Word Mode Code.

Parity. The last bit (bit time 20), is the word parity bit. Only odd parity shall be used.

1.3.2 Message Formats, Validation, and Timing

The primary purpose of the data bus is to provide a common medium for the exchange of data between systems. The exchange of data is based upon message transmissions. The standard defines 10 types of message transmission formats. All of these formats are based upon the three word types just defined. The 10 message formats are shown in Figures 1.6 and 1.7. The message formats have been divided into two groups. These are referred to within the standard as the "information transfer formats" (Figure 1.6) and the "broadcast information transfer formats" (Figure 1.7).

The information transfer formats are based upon the command/response philosophy that all error-free transmissions received by a remote terminal be followed by the transmission of a status word from the terminal to the bus controller. This handshaking principle validates the receipt of the message by the remote terminal.

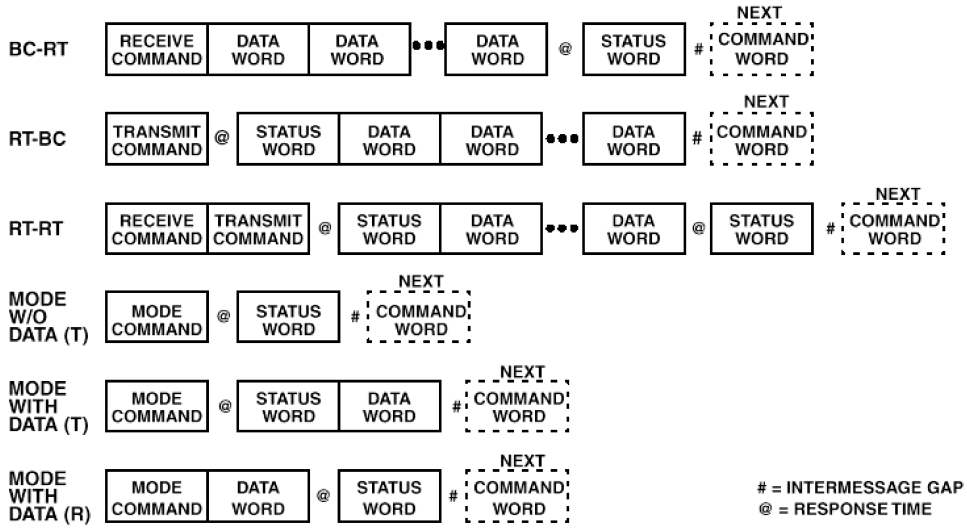

FIGURE 1.6 Information transfer formats.

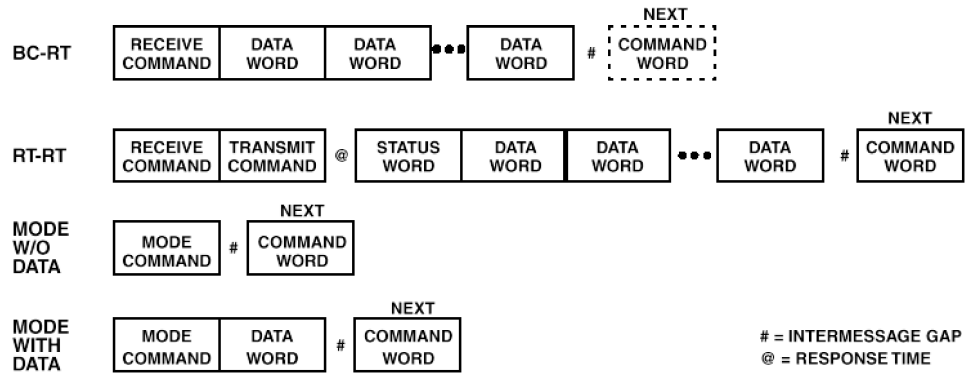

FIGURE 1.7 Broadcast information transfer formats.

Broadcast messages are transmitted to multiple remote terminals at the same time. As such, the terminals suppress the transmission of their status words (not doing so would have multiple boxes trying to talk at the same time and thereby "jam" the bus). In order for the bus controller to determine if a terminal received the message, a polling sequence to each terminal must be initiated to collect the status words.

Each of the message formats is summarized in the following subsections.

1.3.2.1 Bus Controller to Remote Terminal

The bus controller to remote terminal (BC-RT) message is referred to as the receive command since the remote terminal is going to receive data. The bus controller outputs a command word to the terminal defining the subaddress of the data and the number of data words it is sending. Immediately (without any gap in the transmission), the number of data words specified in the command word are sent.

The remote terminal upon validating the command word and all of the data words will issue its status word within the response time requirements (maximum of 12 μsec).

The remote terminal must be capable of processing the next command that the bus controller issues. Therefore the remote terminal has approximately 56 μsec (status word response time 12 μsec, plus status word transmit time 20 μsec, plus intermessage gap minimum 4 μsec, plus command word transmit time 20 μsec, to either pass the data to the subsystem or buffer the data.

1.3.2.2 Remote Terminal to Bus Controller

The remote terminal to bus controller (RT-BC) message is referred to as a transmit command. The bus controller issues only a transmit command word to the remote terminal. The terminal, upon validation of the command word, will first transmit its status word followed by the number of data words requested by the command word.

Since the remote terminal does not know the sequence of commands to be sent and does not normally operate upon a command until the command word has been validated, it must be capable of fetching from the subsystem the data required within approximately 28 μsec (the status word response time 12 μsec, plus the status word transmission time 20 μsec, minus some amount of time for message validation and transmission delays through the encoder and transceiver).

1.3.2.3 Remote Terminal to Remote Terminal

The remote terminal to remote terminal (RT-RT) command is provided to allow a terminal (the data source) to transfer data directly to another terminal (the data sink) without going through the bus controller. The bus controller may, however, collect and use the data.

The bus controller first issues a command word to the receiving terminal immediately followed by a command word to the transmitting terminal. The receiving terminal is expecting data, but instead of data after the command word it sees a command sync (the second command word). The receiving terminal ignores this word and waits for a word with a data sync.

The transmitting terminal ignored the first command word (it did not contain its terminal address). The second word was addressed to it, so it will process the command as an RT-BC command as described above by transmitting its status word and the required data words.

The receiving terminal, having ignored the second command word, again sees a command (status) sync on the next word and waits further. The next word (the first data word sent) now has a data sync and the receiving remote terminal starts collecting data. After receipt of all of the data words (and validating), the terminal transmits its status word.

1.3.2.3.1 RT-RT Validation

There are several things that the receiving remote terminal of an RT-RT message should do. First, Notice 2 requires that the terminal time out in 54 to 60 μsec after receipt of the command word. This is required since if the transmitting remote terminal did not validate its command word (and no transmission occurred) then the receiving terminal will not collect data from some new message. This could occur if the next message is either a transmit or receive message, where the terminal ignores all words with a

command/status sync and would start collecting data words beginning with the first data sync. If the same number of data words were being transferred in the follow-on message and the terminal did not test the command/status word contents, then the potential exists for the terminal to collect erroneous data.

The other function that the receiving terminal should do, but is not required by the standard, is to capture the second command word and the first transmitted data word. The terminal could compare the terminal address fields of both words to insure that the terminal doing the transmitting was the one commanded to transmit. This would allow the terminal to provide a level of protection for its data and subsystem.

1.3.2.4 Mode Command Formats

Three mode command formats are provided for. This allows for mode commands with no data words and for the mode commands with one data word (either transmitted or received). The status/data sequencing is as described for the BC-RT or RT-BC messages except that the data word count is either one or zero. Mode codes and their use are described later.

1.3.2.5 Broadcast Information Transfer Formats

The broadcast information transfer formats, as shown in Figure 1.8, are identical to the nonbroadcast formats described above with the following two exceptions. First, the bus controller issues commands to terminal address 31 (11111) which is reserved for this function. And secondly, the remote terminals receiving the messages (those which implement the broadcast option) suppress the transmission of their status word.

The broadcast option can be used with the message formats in which the remote terminal receives data. Obviously, multiple terminals cannot transmit data at the same time, so the RT-BC transfer format and the transmit mode code with data format cannot be used. The broadcast RT-RT allows the bus controller to instruct all remote terminals to receive and then instructs one terminal to transmit, thereby allowing a single subsystem to transfer its data directly to multiple users.

Notice 2 allows the bus controller to only use broadcast commands with mode codes (see Broadcast Mode Codes). Remote terminals are allowed to implement this option for all broadcast message formats. The Notice further states that the terminal must differentiate the subaddresses between broadcast and nonbroadcast messages (see Subaddress Utilization).

1.3.2.6 Command and Message Validation

The remote terminal must validate the command word and all data words received as part of the message. The criteria for a valid command word are that the: word begins with a valid command sync, valid terminal address (matches the assigned address of the terminal or the broadcast address if implemented), all bits are in a valid Manchester code, there are 16 information field bits, and there is a valid parity bit (odd). The criteria for a data word are the same except a valid data sync is required and the terminal address field is not tested. If a command word fails to meet the criteria, the command is ignored. After the command has been validated, and a data word fails to meet the criteria, then the terminal shall set the Message Error bit in the status word and suppress the transmission of the status word. Any single error within a message shall invalidate the entire message and the data shall not be used.

1.3.2.7 Illegal Commands

The standard allows remote terminals the option of monitoring for Illegal Commands. An Illegal Command is one that meets the valid criteria for a command word, but is a command (message) that is not implemented by the terminal. An example is if a terminal only outputs 04 data words to subaddress 01 and a command word was received by the terminal that requested it to transmit 06 data words from subaddress 03, then this command, while still a valid command, could be considered by the terminal as illegal. The standard only states that the bus controller shall not issue illegal or invalid commands.

The standard provides the terminal designer with two options. First, the terminal can respond to all commands as usual (this is referred to as “responding in form”). The data received is typically placed in a series of memory locations which are not accessible by the subsystem or applications programs.

This is typically referred to as the "bit bucket." All invalid commands are placed into the same bit bucket. For invalid transmit commands, the data transmitted is read from the bit bucket. Remember, the bus controller is not supposed to send these invalid commands.

The second option is for the terminal to monitor for Illegal Commands. For most terminal designs, this is as simple as a look-up table with the T/R bit, subaddress, and word count fields supplying the address and the output being a single bit that indicates if the command is valid or not. If a terminal implements Illegal Command detection and an illegal command is received, the terminal sets the Message Error bit in the status word and responds with the status word.

1.3.2.8 Terminal Response Time

The standard states that a remote terminal, upon validation of a transmit command word or a receive message (command word and all data words) shall transmit its status word to the bus controller. The response time is the amount of time the terminal has to transmit its status word. To allow for accurate measurements, the time frame is measured from the mid-crossing of the parity bit of the command word to the mid-crossing of the sync field of the status word. The minimum time is 4.0 μsec, the maximum time is 12.0 μsec. However, the actual amount of "dead time" on the bus is 2 to 10 μsec since half of the parity and sync waveforms are being transmitted during the measured time frame.

The standard also specifies that the bus controller must wait a minimum of 14.0 μsec for a status word response before determining that a terminal has failed to respond. In applications where long data buses are used or where other special conditions exist, it may be necessary to extend this time to 20.0 μsec or greater.

1.3.2.9 Intermessage Gap

The bus controller must provide for a minimum of 4.0 μsec between messages. Again, this time frame is measured from the mid-crossing of the parity bit of the last data word or the status word and the mid-crossing of the sync field of the next command word. The actual amount of "dead time" on the bus is 2 μsec since half of the parity and sync waveforms are being transmitted during the measured time frame.

The amount of time required by the bus controller to issue the next command is a function of the controller type (e.g., word, message, or frame). The gap typically associated with word controllers is between 40 and 100 μsec. Message controllers typically can issue commands with a gap of 10 to 30 μsec. But frame controllers are capable of issuing commands at the 4-μsec rate and often must require a time delay to slow them down.

1.3.2.10 Superseding Commands

A remote terminal must always be capable of receiving a new command. This may occur while operating on a command on bus A and after the minimum intermessage gap, a new command appears, or if operating on bus A and a new command appears on bus B. This is referred to as a Superseding Command. A second valid command (the new command) shall cause the terminal to stop operating on the first command and start on the second. For dual redundant applications, this requirement implies that all terminals must, as a minimum, have two receivers, two decoders, and two sets of command word validation logic.

1.3.3 Mode Codes

Mode codes are defined by the standard to provide the bus controller with data bus management and error handling/recovery capability. The mode codes are divided into two groups: with and without data words. The data words that are associated with the mode codes, and only one word per mode code is allowed, contains information pertinent to the control of the bus and do not generally contain information required by the subsystem (the exception may be the Synchronize with Data Word Mode Code). The mode codes are defined by bit times 15–19 of the command word. The most significant bit (bit 15) can be used to differentiate between the two mode code groups. When a data word is associated with

the mode code, the T/R bit determines if the data word is transmitted or received by the remote terminal. The mode codes are listed in Table 1.4.

TABLE 1.4 Mode Code

T/R	Mode Code	Function	Data Word	Broadcast
1	00000	Dynamic Bus Control	No	No
1	00001	Synchronize	No	Yes
1	00010	Transmit Status Word	No	No
1	00011	Initiate Self-Test	No	Yes
1	00100	Transmitter Shutdown	No	Yes
1	00101	Override Transmitter Shutdown	No	Yes
1	00110	Inhibit Terminal Flag Bit	No	Yes
1	00111	Override Inhibit Terminal Flag Bit	No	Yes
1	01000	Reset	No	Yes
1	01001	RESERVED	No	TBD
1	•	•	No	•
1	•	•	No	•
1	01111	RESERVED	No	TBD
1	10000	Transmit Vector Word	Yes	No
0	10001	Synchronize	Yes	Yes
1	10010	Transmit Last Command Word	Yes	No
1	10011	Transmit BIT Word	Yes	No
0	10100	Selected Transmitter Shutdown	Yes	Yes
0	10101	Override Selected Transmitter Shutdown	Yes	Yes
1/0	10110	RESERVED	Yes	TBD
	•	•	Yes	•
	•	•	Yes	•
1/0	11111	RESERVED	Yes	TBD

1.3.3.1 Mode Code Identifier

The mode code identifier is contained in bits 10–14 of the command word. When this field is either 00000 or 11111 then the contents of bits 15–19 of the command word are to be decoded as a mode code. Two mode code identifiers are provided such that the system can utilize the Instrumentation bit if desired. The two mode code identifiers shall not convey different information.

1.3.3.2 Mode Code Functions

The following defines the functionality of each of the mode codes.

Dynamic Bus Control. The Dynamic Bus Control Mode Code is used to provide for the passing of the control of the data bus between terminals, thus providing a "round robin" type of control. Using this methodology, each terminal is responsible for collecting the data it needs from all the other terminals. When it is done collecting, it passes control to the next terminal in line (based on some predefined sequence). This allows the applications program (the end user of the data) to collect the data when it needs it, always insuring that the data collected is from the latest source sample and has not been sitting around in a buffer waiting to be used.

Notices 1 and 2 to the standard forbid the use of Dynamic Bus Control for Air Force applications. This is due to the problems and concerns of what may occur when a terminal, that has passed the control, is unable to perform or does not properly forward the control to the next terminal, thereby forcing the condition of no terminal being in control and having to reestablish control by some terminal. The potential amount of time required to reestablish control could have disastrous effects upon the system (i.e., especially a flight control system).

A remote terminal that is capable of performing as the bus control should be capable of setting the Dynamic Bus Control Acceptance Bit in the terminal's Status Word to logic "1" when it receives the mode code command. Typically, the logic associated with the setting of this bit is based on the subsystem's

(computer's) ability to pass some level of confidence test. If the confidence test passes, then the bit is set and the status word is transmitted when the terminal receives the mode command, thereby saying that it will assume the role of bus controller.

The bus controller can only issue the Dynamic Bus Control mode command to one remote terminal at a time. The command obviously is only issued to terminals that are capable of performing as a bus controller. Upon transmitting the command, the bus controller must check the terminal's status word to determine if the Dynamic Bus Control Acceptance Bit is set. If set, the bus controller ceases to function as the controller and becomes either a remote terminal or a bus monitor. If the bit in the status word is not set, the remote terminal which was issued the command is not capable of becoming the bus controller; the current controller must either remain the bus controller or attempt to pass the control to some other terminal.

Synchronize. The synchronize mode code is used to establish some form of timing between two or more terminals. This mode code does not use a data word, therefore the receipt of this command by a terminal must cause some predefined event to occur. Some examples of this event may be the clearing, incrementing, or presetting of a counter; the toggling of an output signal; or the calling of some software routine. Typically, this command is used to time correlate a function such as the sampling of navigation data (i.e., present position, rates, etc.) for flight controls or targeting/fire control systems. Other uses have been for the bus controller to "sync" the back-up controllers (or monitors) to the beginning of a major/minor frame processing.

When a remote terminal receives the Synchronize Mode Command, it should perform its predefined function. For a bus controller, the issuance of the command is all that is needed. The terminals status word only indicates that the message was received, not that the "sync" function was performed.

Transmit Status Word. This is one of the two commands that does not cause the remote terminal to reset or clear its status word. Upon receipt of this command, the remote terminal transmits the status word that was associated with the previous message, not the status word of the mode code message.

The bus controller uses this command for control and error management of the data bus. If the remote terminal had detected an error in the message and suppressed its status word, then the bus controller can issue this command to the remote terminal to determine if indeed the nonresponse was due to an error. As this command does not clear the status word from the previous message, a detected error by the remote terminal in a previous message would be indicated by having the Message Error bit set in the status word.

The bus controller also uses this command when "polling." If a terminal does not have periodic messages, the RT can indicate when it needs communications by setting the Service Request bit in the status word. The bus controller, by requesting the terminal to transmit only its status word, can determine if the terminal is in need of servicing and can subsequently issue the necessary commands. This "polling" methodology has the potential of reducing the amount of bus traffic by eliminating the transmission of unnecessary words.

Another use of this command is when broadcast message formats are used. As all of the remote terminals will suppress their status words, "polling" each terminal for its status word would reveal whether the terminal received the message by having its Broadcast Command Received bit set.

Initiate Self-Test. This command, when received by the remote terminal, shall cause the remote terminal to enter into its self-test. This command is normally used as a ground-based maintenance function, as part of the system power-on tests, or in flight as part of a fault recovery routine. Note that this test is only for the remote terminal, not the subsystem.

In earlier applications, some remote terminals, upon receipt of this command, would enter self-test and go "offline" for long periods of time. Notice 2, in an effort to control the amount of time that a terminal could be "offline," limited the test time to 100.0 μsec following the transmission of the status word by the remote terminal.

While a terminal is performing its self-test, it may respond to a valid command in the following ways: (a) no response on either bus ("off-line"); (b) transmit only the status word with the Busy bit set; or

(c) normal response. The remote terminal may, upon receipt of a valid command received after this mode code, terminate its self-test. As a subsequent command could abort the self-test, the bus controller, after issuing this command, should suspend transmissions to the terminal for the specified amount of time (either a time specified for the remote terminal or the maximum time of 100.0 μsec).

Transmitter Shutdown. This command is used by the bus controller in the management of the bus. In the event that a terminal's transmitter continuously transmits, this command provides for a mechanism to turn the transmitter off. This command is for dual redundant standby applications only.

Upon receipt of this command, the remote terminal shuts down (i.e., turns off) the transmitter associated with the opposite data bus. That is to say if a terminals transmitter is babbling on the A bus, the bus controller would send this command to the terminal on the B bus (a command on the A bus would not be received by the terminal).

Override Transmitter Shutdown. This command is the complement of the previous one in that it provides a mechanism to turn on a transmitter that had previously been turned off. When the remote terminal receives this command, it shall set its control logic such that the transmitter associated with the opposite bus be allowed to transmit when a valid command is received on the opposite bus. The only other command that can enable the transmitter is the Reset Remote Terminal Mode Command.

Inhibit Terminal Flag. This command provides for the control of the Terminal Flag bit in a terminal's status word. The Terminal Flag bit indicates that there is a error within the remote terminal hardware and that the data being transmitted or the data received may be in error. However, the fault within the terminal may not have any effect upon the quality of the data, and the bus controller may elect to continue with the transmissions knowing a fault exists.

The remote terminal receiving this command shall set its Terminal Flag bit to logic "0" regardless of the true state of this signal. The standard does not state that the built-in-test that controls this bit be halted, but only the results be negated to "0."

Override Inhibit Terminal Flag. This command is the complement of the previous one in that it provides a mechanism to turn on the reporting of the Terminal Flag bit. When the remote terminal receives this command, it shall set its control logic such that the Terminal Flag bit is properly reported based upon the results of the terminal's built-in-test functions. The only other command that can enable the response of the Terminal Flag bit is the Reset Remote Terminal Mode Command.

Reset Remote Terminal. This command, when received by the remote terminal, shall cause the terminal electronics to reset to its power-up state. This means that if a transmitter had been disabled or the Terminal Flag bit inhibited, these functions would be reset as if the terminal had just powered up. Again, remember that the reset applies only to the remote terminal electronics and not to the entire box.

Notice 2 restricts the amount of time that a remote terminal can take to reset its electronics. After transmission of its status word, the remote terminal shall reset within 5.0 μsec. While a terminal is resetting, it may respond to a valid command in the following ways: (a) no response on either bus ("offline"); (b) transmit only the status word with the Busy bit set; or (c) normal response. The remote terminal may, upon receipt of a valid command received after this mode code, terminate its reset function. As a subsequent command could abort the reset, the bus controller, after issuing this command, should suspend transmissions to the terminal for the specified amount of time (either a time specified for the remote terminal or the maximum time of 5.0 μsec).

Transmit Vector Word. This command shall cause the remote terminal to transmit a data word referred to as the vector word. The vector word shall identify to the bus controller service request information relating to the message needs of the remote terminal. While not required, this mode code is often tied to the Service Request bit in the Status Word. As indicated, the contents of the data word inform the bus controller of messages that need to be sent.

The bus controller also uses this command when "polling." Though typically used in conjunction with the Service Request bit in the status word, wherein the bus controller requests only the status word (Transmit Status Word Mode Code) and upon seeing the Service Request bit set would then issue the Transmit Vector Word Mode Code, the bus controller can always ask for the Vector Word (always getting the status word anyway) and reduce the amount of time required to respond to the terminal's request.

Synchronize with Data Word. The purpose of this synchronize command is the same as the synchronize without data word, except this mode code provides a data word to provide additional information to the remote terminal. The contents of the data word are left to the imagination of the user. Examples from "real world" applications have used this word to provide the remote terminal with a counter or clock value; to provide a backup controller with a frame identification number (minor frame or cycle number); and to provide a terminal with a new base address pointer used in extending the subaddress capability.

Transmit Last Command Word. This is one of the two commands that does not cause the remote terminal to reset or clear its status word. Upon receipt of this command, the remote terminal transmits the status word that was associated with the previous message and the Last Command Word (valid) that it received.

The bus controller uses this command for control and error management of the data bus. When a remote terminal is not responding properly, then the bus controller can determine the last valid command the terminal received and can re-issue subsequent messages as required.

Transmit BIT Word. This mode command is used to provide detail with regards to the Built-in-Test (BIT) status of the remote terminal. Its contents shall provide information regarding the remote terminal only (remember the definition) and not the subsystem.

While most applications associate this command with the Initiate Self Test Mode Code, the standard requires no such association. Typical use is to issue the Initiate Self Test Mode Code, allow the required amount of time for the terminal to complete its tests, and then issue the Transmit BIT Word Mode Code to collect the results of the test. Other applications have updated the BIT word on a periodic rate based on the results of a continuous background test (e.g., as a data wraparound test performed with every data transmission). This word can then be transmitted to the bus controller, upon request, without having to initiate the test and then wait for the test to be completed. The contents of the data word are left to the terminal designer.

Selected Transmitter Shutdown. Like the Transmitter Shutdown Mode Code, this mode code is used to turn off a babbling transmitter. The difference between the two mode codes is that this mode code has a data word associated with it. The contents of the data word specifies which data bus (transmitter) to shutdown. This command is used in systems which provide more than dual redundancy.

Override Selected Transmitter Shutdown. This command is the complement of the previous one in that it provides a mechanism to turn on a transmitter that had previously been turned off. When the remote terminal receives this command, the data word specifies which data bus (transmitter) shall set its control logic such that the transmitter associated with that bus be allowed to transmit when a valid command is received on that bus. The only other command that can enable the selected transmitter is the Reset Remote Terminal Mode Command.

Reserved Mode Codes. As can be seen from Table 1.4, there are several bit combinations that are set aside as reserved. It was the intent of the standard that these be reserved for future growth. It should also be noticed from the table that certain bit combinations are not listed. The standard allows the remote terminal to respond to these reserved and "undefined" mode codes in the following manner: set the message error bit and respond (see Illegal Commands); or respond in form. The designer of terminal hardware or a multiplex system is forbidden to use the reserved mode codes for any purpose.

1.3.3.3 Required Mode Codes

Notice 2 to the standard requires that all remote terminals implement the following four mode codes: Transmit Status Word, Transmitter Shutdown, Override Transmitter Shutdown, and Reset Remote Terminal. This requirement was levied so as to provide the multiplex system designer and the bus controller with a minimum set of commands for managing the multiplex system. Note that the above requirement was placed on the remote terminal. Notice 2 also requires that a bus controller be capable of implementing all of the mode codes, however, for Air Force applications, the Dynamic Bus Control Mode Code shall never be used.

1.3.3.4 Broadcast Mode Codes

Notice 2 to the standard allows the broadcast of mode codes (see Table 1.4). The use of the broadcast option can be of great assistance in the areas of terminal synchronization. Ground maintenance and troubleshooting can take advantage of broadcast Reset Remote Terminal or Initiate Self, but these two commands can have disastrous effects if used while in flight. The designer must provide checks to insure that commands such as these are not issued by the bus controller or operated upon by a remote terminal when certain conditions exists (e.g., in flight).

1.4 Systems-Level Issues

The standard provides very little guidance in how it is applied. Lessons learned from "real world" applications have led to design guides, application notes, and handbooks that provide guidance. This section will attempt to answer some of the systems-level questions and identify implied requirements that, while not specifically called out in the standard, are required nonetheless.

1.4.1 Subaddress Utilization

The standard provides no guidance on how to use the subaddresses. The assignment of subaddresses and their functions (the data content) is left to the user. Most designers automatically start assigning subaddresses with 01 and count upwards. If the Instrumentation bit is going to be used, then the subaddresses must start at 16.

The standard also requires that normal subaddresses be separated from broadcast subaddresses. If the broadcast option is implemented, then an additional memory block is required to receive broadcast commands.

1.4.1.1 Extended Subaddressing

The number of subaddresses that a terminal has is limited to 60 (30 transmit and 30 receive). Therefore, the number of unique data words available to a terminal is 1920 (60×32). For earlier applications, where data being transferred were analog sensor data and switch settings, this was more than sufficient. However, in some of today's applications, in which digital computers exchanging data, or for a video sensor passing digitized video data, the number of words is too limited.

Most terminal designs establish a block of memory for use by the 1553 interface circuitry. This block contains a address start pointer and then the memory is offset by the subaddress number and the word count number to arrive at a particular memory address.

A methodology of extending the range of the subaddresses has been successfully utilized. This method uses either a dedicated subaddress and data word, or makes use of the synchronize with data word mode code. The data word associated with either of these contains an address pointer which is used to reestablish the starting address of the memory block. The changing of the blocks is controlled by the bus controller and can be done based on numerous functions. Examples are operational modes, wherein one block is used for startup messages, a different block for take-off and landing, a different block for navigation and cruise, a different block for mission functions (i.e., attack or evade modes), and a different block for maintenance functions.

Another example is that the changing of the start address could also be associated with minor frame cycles. Eight minor frames could have a separate memory block for each frame. The bus controller could synchronize frames and change memory pointers at the beginning of each new minor frame.

For computers exchanging large amounts of data (e.g., GPS Almanac Tables) or for computers that receive program loads via the data bus at power-up, the bus controller could set the pointers at the beginning of a message block, send 30, 32-word messages, move the memory pointer to the last location in the remote terminals memory that received data, then send the next block of 30, 32-word messages, continuing this cycle until the memory is loaded. The use is left to the designer.

1.4.2 Data Wraparound

Notice 2 to the standard does require that the terminal is able to perform a data wraparound and subaddress 30 is suggested for this function. Data wraparound provides the bus controller with a methodology of testing the data bus from its internal circuitry, through the bus media, to the terminal's internal circuitry. This is done by the bus controller sending the remote terminal a message block and then commanding the terminal to send it back. The bus controller can then compare the sent data with that received to determine the state of the data link. There are no special requirements upon the bit patterns of the data being transferred.

The only design requirements are placed upon the remote terminal. These are that the terminal, for the data wraparound function, be capable of sending the number of data words equal to the largest number of data words sent for any transmit command. This means that if a terminal maximum data transmission is only four data words, it need only provide for four data words in its data wraparound function.

The other requirement is that the remote terminal need only hold the data until the next message. The normal sequence is for the bus controller to send the data, then in the next message it asks for it back. If another message is received by the remote terminal before the bus controller requests the data, the terminal can discard the data from the wraparound message and operate on the new command.

1.4.3 Data Buffering

The standard specifies that the any error within a message shall invalidate the entire message. This implies that the remote terminal must store the data within a message buffer until the last data word has been received and validated before allowing the subsystem access to the data. To insure that the subsystem always has the last message of valid data received to work with would require the remote terminal to, as a minimum, double buffer the received data.

There are several methods to accomplish this in hardware. One method is for the terminal electronics to contain a First-In First-Out (FIFO) memory that stores the data as it is received. Upon validation of the last data word, the terminal's subsystem interface logic will move the contents of the FIFO into memory accessible by the subsystem. If an error occurred during the message, the FIFO is reset.

A second method establishes two memory blocks for each message in common memory. The subsystem is directed to read from one block (block A) while the terminal electronics writes to the other (Block B). Upon receipt of a valid message, the terminal will switch pointers, indicating that the subsystem is to read from the new memory block (block B) while the terminal will now write to block B. If an error occurs within the message, the memory blocks are not switched.

Some of the "off-the-shelf" components available provide for data buffering. Most provide for double buffering, while some provided for multilevels of buffering.

1.4.4 Variable Message Blocks

Remote terminals should be able to transmit any subset of any message. This means that if a terminal has a transmit message at subaddress 04 of 30 data words, it should be capable of transmitting any number of those data words (01–30) if so commanded by the bus controller. The order in which the subset is transmitted should be the same as if the entire message is being transmitted, that being the contents of data word 01 is the same regardless of the word count.

Terminals which implement Illegal Command detection should not consider subsets of a message as illegal. That is to say, if in our example above a command is received for 10 data words, this should not be illegal. But, if a command is received for 32 data words, this would be considered as an illegal command.

1.4.5 Sample Consistency

When transmitting data, the remote terminal needs to ensure that each message transmitted is of the same sample set and contains mutually consistent data. Multiple words used to transfer multiple precision parameters or functionally related data must of the same sampling.

If a terminal is transmitting pitch, roll, and yaw rates, and while transmitting the subsystem updates these data in memory, but this occurs after pitch and roll had been read by the terminal's electronics, then the yaw rate transmitted would be of a different sample set. Having data from different sample rates could have undesirable effects on the user of the data.

This implies that the terminal must provide some level of buffering (the reverse of what was described above) or some level of control logic to block the subsystem from updating data while being read by the remote terminal.

1.4.6 Data Validation

The standard tightly defines the criteria for the validation of a message. All words must meet certain checks (i.e., valid sync, Manchester encoding, number of bits, odd parity, etc.) in order for each word and each message to be valid. But what about the contents of the data word? MIL-STD-1553 provides the checks to insure the quality of the data transmission from terminal to terminal, sort of a "data in equals data out," but is not responsible for the validation tests of the data itself. This is not the responsibility of the 1553 terminal electronics, but of the subsystem. If bad data are sent, then "garbage in equals garbage out." But the standard does not prevent the user from providing additional levels of protection. The same techniques used in digital computer interfaces (i.e., disk drives, serial interfaces, etc.) can be applied to 1553. These techniques include checksums, CRC words, and error detection/correction codes. Section 80 of MIL-HDBK-1553A which covers data word formats even offers some examples of these techniques.

But what about using the simple indicators embedded within the standard. Each remote terminal provides a status word — indicating not only the health of the remote terminal's electronics, but also that of the subsystem. However, in most designs, the status word is kept within the terminal electronics and not passed to the subsystems. In some "off-the-shelf" components, the status word is not even available to be sent to the subsystem. But two bits from the status word should be made available to the subsystem and the user of the data for further determination as to the validity of the data. These are the Subsystem Flag and the Terminal Flag bits.

1.4.7 Major and Minor Frame Timing

The standard specifies the composition of the words (command, data, and status) and the messages (information formats and broadcast formats). It provides a series of management messages (mode codes), but it does not provide any guidance on how to apply these within a system. This is left to the imagination of the user.

Remote terminals, based upon the contents of their data, will typically state how often data are collected and the fastest rate they should be outputted. For input data, the terminal will often state how often it needs certain data to either perform its job or maintain a certain level of accuracy. The rates are referred to as the transmission and update rates. It is the system designer's job to examine the data needs of all of the systems and determine when data are transferred from whom to whom. These data are subdivided into periodic messages — those which must be transferred at some fixed rate, and aperiodic messages, those which are typically either event driven (i.e., the operator pushes a button) or data driven (i.e., a value is now within range).

A major frame is defined such that all periodic messages are transferred at least once. This is therefore defined by the message with the slowest transmission rate. Typical major frame rates used in today's applications vary from 40 to 640 μsec. There are some systems that have major frame rates in the 1- to 5-sec range, but these are the exceptions, not the norm. Minor frames are then established to meet the requirements of the higher update rate messages.

The sequence of messages within a minor frame is again left undefined. There are two methodologies that are predominately used. In the first method, the bus controller starts the frame with the transmission of all of the periodic messages (transmit and receive) to be transferred in that minor frame. At the end of the periodic messages, the bus controller is either finished (resulting in dead bus time — no transmissions) until the beginning of the next frame, or the bus controller can use this time to transfer aperiodic messages, error handling messages, or transfer data to the back-up bus controller(s).

In the second method (typically used in a centralized processing architecture), the bus controller issues all periodic and aperiodic transmit messages (collects the data), then processes the data (possibly using dead time during this processing), and then issues all the receive messages (outputting the results of the processing). Both methods have been used successfully.

1.4.8 Error Processing

The amount and level of error processing is typically left to the systems designer but may be driven by the performance requirements of the system. Error processing is typically only afforded to critical messages, wherein the noncritical messages just await the next normal transmission cycle. If a data bus is 60% loaded and each message received an error, the error processing would exceed 100% of available time and thereby cause problems within the system.

Error processing is again a function of the level of sophistication of the bus controller. Some controllers (typically message or frame controllers) can automatically perform some degree of error processing. This usually is limited to a retransmission of the message either once on the same bus or once on the opposite bus. Should the retried message also fail, the bus controller software is informed of the problem. The message may then be retried at the end of the normal message list for the minor frame.

If the error still persists, then it may be necessary to stop communicating with the terminal, especially if the bus controller is spending a large amount of time performing error processing. Some systems will try to communicate with a terminal for a predefined number of times on each bus. After this, all messages to the terminal are removed from the minor frame lists, and substituted with a single transmit status word mode code.

An analysis should be performed on the critical messages to determine the effects upon the system if they are not transmitted or the effects of data latency if they are delayed to the end of the frame.

1.5 Testing

The testing of a MIL-STD-1553 terminal or system is not a trivial task. There are a large number of options available to the designer including message formats, mode commands, status word bits, and coupling methodology. In addition, history has shown that different component manufacturers and designers have made different interpretations regarding the standard, thereby introducing products that implement the same function quite differently.

For years, the Air Force provided for the testing of MIL- STD-1553 terminals and components. Today this testing is the responsibility of industry. The Society of Automotive Engineers (SAE), in conjunction with the government, has developed a series of Test Plans for all 1553 elements. These Test Plans are listed in Table 1.6.

TABLE 1.6 SAE 1553 Test Plans

AS-4111	Remote Terminal Validation Test Plan
AS-4112	Remote Terminal Production Test Plan
AS-4113	Bus Controller Validation Test Plan
AS-4114	Bus Controller Production Test Plan
AS-4115	Data Bus System Test Plan
AS-4116	Bus Monitor Test Plan
AS-4117	Bus Components Test Plan

Further Information

In addition to the SAE Test Plans listed in Table 1.6, there are other documents that can provide a great deal of insight and assistance in designing with MIL-STD-1553:

MIL-STD-1553B Digital Time Division Command/Response Multiplex Data Bus
MIL-HDBK-1553A Multiplex Applications Handbook
SAE AS-15531 Digital Time Division Command/Response Multiplex Data Bus
SAE AS-15532 Standard Data Word Formats
SAE AS-12 Multiplex Systems Integration Handbook
SAE AS-19 MIL-STD-1553 Protocol Reorganized
DDC 1553 Designers Guide
UTMC 1553 Handbook

And lastly, there is the SAE 1553 Users Group. This is a collection of industry and military experts in 1553 who provide an open forum for information exchange, and provide guidance and interpretations/clarifications with regard to the standard. This group meets twice a year as part of the SAE Avionics Systems Division conferences.

Daniel A. Martinec "ARINC 429"
The Avionics Handbook
Ed. Cary R. Spitzer
Boca Raton, CRC Press LLC. 2001