

IEEE Standard for Module Test and Maintenance Bus (MTM-Bus) Protocol

Sponsor

**Test Technology Standards Committee
of the
IEEE Computer Society**

Approved August 14, 1995

IEEE Standards Board

Abstract: This Standard specifies a serial, backplane, test and maintenance bus (MTM-Bus) that can be used to integrate modules from different design teams or vendors into testable and maintainable subsystems. Physical, link, and command layers are specified. Standard interface protocol and commands can be used to provide the basic test and maintenance features needed for a module as well as access to on-module assets (memory, peripherals, etc.) and IEEE Std 1149.1 boundary-scan. Standard commands and functions support fault isolation to individual modules and test of backplane interconnect between modules.

Keywords: backplane bus, boundary scan, built-in self-test, maintenance, MTM-Bus, subsystems, system diagnostics, system test, testing

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

Copyright © 1996 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1996. Printed in the United States of America.

ISBN 1-55937-558-2

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

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason IEEE and the members of its technical committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

IEEE Standards documents may involve the use of patented technology. Their approval by the Institute of Electrical and Electronics Engineers does not mean that using such technology for the purpose of conforming to such standards is authorized by the patent owner. It is the obligation of the user of such technology to obtain all necessary permissions.

Introduction

[This introduction is not part of IEEE Std 1149.5-1995, IEEE Standard for Module Test and Maintenance Bus (MTM-Bus) Protocol.]

The purpose for developing this Standard is to provide a cost-effective and standardized backplane module test and maintenance interface that can be used to integrate testable modules from different design teams or vendors into testable and maintainable subsystems. Standard interface protocol and commands can be used to provide the basic test and maintenance features needed for a module as well as access to on-module assets (memory, peripherals, etc.) and IEEE Std 1149.1 boundary-scan. Standard commands and functions support fault isolation to individual modules and test of backplane interconnect between modules.

The origins of the protocol for this Standard date back to the United States Department of Defense Very High Speed Integrated Circuits (VHSIC) program in 1986. In that program there was a need to develop a test and maintenance architecture that would be common between the three contractors (Honeywell, IBM, and TRW). This test and maintenance architecture included a backplane test and maintenance bus as well as a chip-level test bus. The backplane test and maintenance bus was further developed for use in several advanced avionics programs. Following the completion of these applications to avionics, significant interest in developing an industrial standard based on this test and maintenance bus were registered. The level of interest was sufficient to initiate the development of a backplane test and maintenance protocol standard as a sibling to the other test-related buses that were being developed in the IEEE 1149 group of standards (1990). The chip-level test architecture of the VHSIC program had previously been folded into IEEE Std 1149.1.

Work on the module test and maintenance bus (referred to herein as “MTM-Bus”) began with the final draft from the previous Department of Defense efforts. This draft was significantly modified to remove all project specific artifacts from previous standardization efforts and generalized to allow applicability to all types of systems. An additional major effort was required to clarify ambiguities so that different vendors of IEEE Std 1149.5 protocol chips could provide parts able to communicate and be interoperable. The total revision and expansion effort took approximately four years and added almost 150 pages to the document. It is interesting to note that the document went through a letter ballot review by more than 100 engineers, who had no substantial comments, except for one negative review (later withdrawn) that disagreed with the fundamental nature of the interface.

In 1995, the VME Standards Organization (VSO) included the IEEE Std 1149.5 MTM-Bus in their VME64 group of standards, which specifies a backplane for embedded systems. Work is continuing on the application of the MTM-Bus in VME64 standard extensions and applications.

At the time of approval of this standard, the P1149.5 MTM-Bus Working Group had the following membership:

Patrick F. McHugh, *Chair*
Rodham E. Tulloss, *Technical Editor*

Dave Heiligenstein
Christopher M. Henwood
Chuck Hudson

Gerry Kendzior
Russell A. Mansfield

Jeffrey W. Lundy
Michel Parot
Gregory D. Young

The following persons were on the balloting committee:

Jacob A. Abraham
Dharma P. Agrawal
John Andrews
Sanjay Bajaj
Behrooz Bandall
Elizabeth Benedict
Dilip K. Bhavsar
Harry Bleeker
Federico Bonzanigo
David W. Browne
Harold W. Carter
R. Chandramouli
John P. Chihorek
Wilton Chiles
John Crownover
R. Dandapani
Wayne Daniel
Steven Dollens
Samuel Duncan
Leo G. Egan
William Eklow
John H. Erickson
Carles Ferrer
Peter Fleming
Andrew Fraser
Walter Geisselhardt
Dong S. Ha
Harry E. Hansen
C. H. Hao
Richard Harms
Werner Hauptmann
Leonard Hause
Charles Hawkins
Alan Hearn
Wei-Cheng Her
Charles Holliday

Kenneth Hoyme
Chuck Hudson
Masaaki Inadachi
Neal Jaarsma
Najmi Jarwala
M. A. Anura P. Jayasumana
Wen-Ben Jone
Carl Kagy
Bozena Kaminska
Jake Karrfalt
Mehdi Katozi
William Keiner
Charles Kime
Steven K. Ladd
David Landis
Bjorn B. Larsen
Luis Larzabal
Robert Ledbetter
Ben Lee
Marc Levitt
Jeffrey W. Lundy
R. A. Mansfield
Ken Mason
Colin Maunder
Thomas J. McGrath
Earl Meiers
Alex Meloni
Yinghua Min
Math Muris
H. Troy Nagle, Jr.
Prawat Nagvajara
James Nagy
Roy H. Nesson
Yoshinori Ninohira
Arnold Nordsieck
Paul Ocampo
Michel Parot

Robin Passow
Stephen Pizzica
Paolo Prinetto
Rochit Rajsuman
Edward Ramirez
Edward P. Ratazzi
Shishpal Rawat
William T. Rhoades
Gordon Robinson
Robert Rolfe
Kenneth Rose
Fred U. Rosenberger
Cayetano Sanchez
Sudha Sarma
William H. Smith
Mani Soma
Gerard N. Stenbakken
James H. Stewart
Jacques Tete
Cihan Tinaztepe
Erwin Trischler
Joseph G. Tront
Rodham E. Tulloss
Jon L. Turino
Louis Y. Ungar
Dirk Van de Lagemaat
Russell J. Wagner
J. Richard Weger
Harry Whittemore
Werner Wolz
Cheng-Wen Wu
Chi Yau
Homayoun Yazdanpanah
Greg D. Young
Farzad Zarrinfar
Guenther Zwiehoff

When the IEEE Standards Board approved this standard on August 14, 1995, it had the following membership:

E. G. "Al" Kiener, *Chair*

Donald C. Loughry, *Vice Chair*

Andrew G. Salem, *Secretary*

Gilles A. Baril
Clyde R. Camp
Joseph A. Cannatelli
Stephen L. Diamond
Harold E. Epstein
Donald C. Fleckenstein
Jay Forster*
Donald N. Heirman

Richard J. Holleman
Jim Isaak
Ben C. Johnson
Sonny Kasturi
Lorraine C. Kevra
Ivor N. Knight
Joseph L. Koepfinger*
D. N. "Jim" Logothetis
L. Bruce McClung

Marco W. Migliaro
Mary Lou Padgett
John W. Pope
Arthur K. Reilly
Gary S. Robinson
Ingo Rusch
Chee Kiow Tan
Leonard L. Tripp

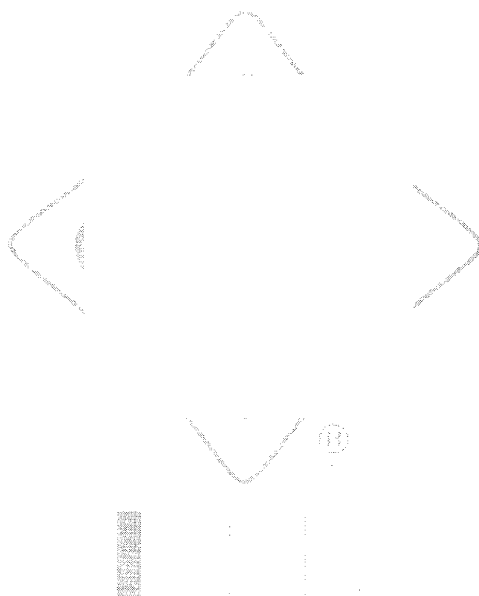
*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
Richard B. Engelman
Robert E. Hebner
Chester C. Taylor

Paula M. Kelty
IEEE Standards Project Editor





Contents

CLAUSE	PAGE
1. Overview.....	1-1
1.1 Scope and objectives	1-1
1.2 Interconnection of modules with MTM-Bus	1-2
1.3 Relationship to other test buses	1-3
1.4 Overview of MTM-Bus operation/protocol	1-3
1.5 MTM-Bus protocol layers	1-5
1.6 Extensions to this Standard	1-6
2. Conventions, definitions, and references	2-1
2.1 Document outline	2-1
2.2 Conventions	2-1
2.3 Definitions	2-2
2.4 References	2-10
3. MTM-Bus architecture.....	3-1
3.1 MTM-Bus signals and interconnection of MTM-Bus modules via these signals	3-1
3.2 General architecture and MTM-Bus mastership	3-2
3.3 S-Module addressing requirements	3-3
4. MTM-Bus Physical Layer requirements.....	4-1
4.1 Physical Layer requirements independent of module type.....	4-1
4.2 M-module Physical Layer requirements	4-2
4.3 S-Module Physical Layer requirements	4-3
4.4 MTM-Bus timing requirements.....	4-4
5. MTM-Bus Link Layer: Packet requirements.....	5-1
5.1 Requirements applicable to all packets	5-1
5.2 HEADER packet requirements.....	5-1
5.3 ACKNOWLEDGE packet requirements.....	5-2
5.4 PACKET COUNT packet requirements	5-3
5.5 DATA packet requirements.....	5-4
5.6 NULL packet requirements	5-5
5.7 Formatting bit strings of more than 16 bits for transmission in DATA packets	5-5
6. MTM-Bus Link Layer: M-module requirements.....	6-1
6.1 MTM-Bus Master Link Layer Controller (M-Controller) requirements.....	6-1
6.2 M-module send and receive parity requirements	6-9
6.3 M-module MPR signal (data transfer control) requirements	6-10
6.4 M-module response to collision errors on MMD and MCTL signals	6-11
6-5 M-module interrupt response requirements.....	6-11

CLAUSE	PAGE
7. MTM-Bus Link Layer: S-module requirements—MTM-Bus Slave Link Layer Controller.....	7-1
7.1 MTM-Bus Slave Link Layer Controller requirements	7-1
7.2 S-module interface logic requirements.....	7-8
8. MTM-Bus Link Layer: S-module requirements—general communications.....	8-1
8.1 S-module send and receive parity requirements	8-1
8.2 S-module MSD signal general requirements	8-1
8.3 S-module MTM-Bus Pause Request (MPR) signal implementation (data transfer control) requirements.....	8-2
9. MTM-Bus Link Layer: S-module requirements—status registers	9-1
9.1 S-module status registers—general requirements	9-1
9.2 Slave Status register	9-2
9.3 Bus Error register	9-7
9.4 Module Status register.....	9-15
9.5 Additional status registers	9-16
10. MTM-Bus Link Layer: S-module interrupt generation	10-1
10.1 Interrupt generation other than immediate response to a State Sequence Error.....	10-1
11. S-module error response requirements	11-1
11.1 S-module error response—general requirements	11-1
11.2 S-module self-test failure response requirements.....	11-3
11.3 Broadcast and multicast errors	11-4
11.4 S-module Data Overrun Error and response requirements.....	11-5
11.5 S-module Parity Error response requirements.....	11-5
11.6 S-module State Sequence Error response requirements	11-6
11.7 MSD signal collision response requirements	11-8
11.8 S-module Data Transfer Port Error response requirements	11-9
11.9 S-module Command Sequence Error response requirements	11-10
11.10 S-module Illegal Command Error response requirements	11-11
11.11 S-module Packet Count Error response requirements.....	11-12
11.12 S-module Command Resource Unavailable Error response requirements	11-13
12. MTM-Bus Message Layer: General requirements	12-1
12.1 MTM-Bus Message Layer general requirements	12-1
13. MTM-Bus Message Layer: M-module requirements	13-1
13.1 M-module PACKET COUNT packet transmission requirements	13-1
13.2 Post-error recovery at packet and message levels	13-1

CLAUSE	PAGE
14. MTM-Bus Message Layer: S-module requirements.....	14-1
14.1 S-module general HEADER packet decode and general command response requirements	14-1
14.2 S-module packet-counting requirements.....	14-2
14.3 Summary of S-module message sequence requirements.....	14-3
15. MTM-Bus Message Layer: Commands—general requirements	15-1
15.1 MTM-Bus commands—general requirements on command codes and command classes	15-1
15.2 Commands execution of which may be terminated upon receipt of another command	15-4
16. MTM-Bus Message Layer: Core class commands (0000000–0011111; 1111111)	16-1
16.1 The Read Status command (0000000)	16-1
16.2 Abort command (0000001)	16-4
16.3 Reset Slave Status command (0000010).....	16-5
16.4 Contend for Bus command (0000011)	16-6
16.5 Multicast Select n commands ($n = 0, 1, 2, 3$) (0000100–0000111)	16-9
16.6 Enable Idle Interrupts command (0001000).....	16-11
16.7 Enable Pause Interrupts command (0001001).....	16-11
16.8 Disable Idle Interrupts command (0001010).....	16-12
16.9 Disable Pause Interrupts command (0001011).....	16-13
16.10 Enable Module Control (EMC) command (0001100).....	16-13
16.11 Data Echo Test command (0001101).....	16-15
16.12 Verify BMR command (0001110)	16-16
16.13 The Initialize Application command (0001111)	16-18
16.14 Disable Module Control command (0010000).....	16-19
16.15 Start command (0010001).....	16-20
16.16 Reserved command (0010010–0011111).....	16-21
16.17 Illegal command (1111111)	16-21
17. MTM-Bus Message Layer: Data Transfer class commands (0100000–0100111)	17-1
17.1 General format requirements for Data Transfer class commands	17-1
17.2 Read Data command (0100000).....	17-2
17.3 Write Data command (0100001).....	17-4
17.4 Read/Write Data command (0100010).....	17-5
17.5 Reserved commands (0100011–0100111)	17-7
18. MTM-Bus Message Layer: Module Initialization and Self-Test (MIST) class commands (0100000–0100111).....	18-1
18.1 General requirements for MIST commands	18-2
18.2 Reset Module With Start-up Built-in Test (SBIT) command (0101000)	18-2
18.3 Reset Module Without SBIT command (0101001).....	18-4
18.4 Module Initiated Built-in Test (Module IBIT) command (0101010).....	18-5
18.5 Reserved commands (0101011–0101111)	18-7

CLAUSE	PAGE
19. MTM-Bus Message Layer: Module I/O Control and Test (MICT) class commands (0110000–0110111).....	19-1
19.1 MICT class commands—general requirements	19-2
19.2 Disable Module I/O command (0110000).....	19-4
19.3 Enable Module I/O command (0110001).....	19-6
19.4 Force Module Outputs command (0110010)	19-7
19.5 Sample Module Inputs commands (0110011–0110101)—general requirements	19-9
19.6 Sample Module No Change command (0110011)	19-11
19.7 Sample Module Don't Care command (0110100)	19-11
19.8 Sample Module With Force command (0110101)	19-12
19.9 Release Module I/O command (0110110)	19-13
19.10 Reserved commands (0110111–1001111)	19-14
20. MTM-Bus Message Layer: Standard Extension class commands (1010000–1011111) and User-Defined class commands (1100000–1111110).....	20-1
20.1 Standard Extension class commands (1010000–1011111).....	20-1
20.2 User-Defined class commands (1100000–1111110).....	20-1
21. Data transfer ports.....	21-1
21.1 Data transfer ports—general requirements.....	21-1
21.2 Port definition and documentation requirements	21-3
21.3 Module/Fault log port(s) ('0000' HEX – '0003' HEX)	21-3
21.4 Test Data Storage ports ('0004' HEX – '0007' HEX)	21-4
21.5 Error/Status register ports ('0008' HEX – '000B' HEX).....	21-5
21.6 MTM-Bus Interface Manufacturer ID port ('000C' HEX)	21-6
21.7 Module Manufacturer port ('000D' HEX)	21-8
21.8 User Identification ports ('000E' HEX – '000F' HEX).....	21-10
21.9 Access to status register backup means—Error/Status Shadow register port ('0080' HEX).....	21-11
21.10 Ports interfacing to IEEE Std 1149.1 boundary-scan ('0010' HEX – '001F' HEX)	21-12
21.11 IEEE Std 1149.1 Interface ports—requirements for the Full TAP Control (FTC) access method.....	21-19
21.12 IEEE Std 1149.1 Interface ports—requirements for the Function-Based Control (FBC) access method.....	21-25
22. MTM-Bus general documentation requirements	22-1
22.1 Documentation requirements.....	22-1

IEEE Standard for Module Test and Maintenance Bus (MTM-Bus) Protocol

1. Overview

1.1 Scope and objectives

1.1.1 Scope

This document standardizes a serial, backplane, test and maintenance bus (referred to herein as “MTM-Bus”), consisting of one or more logic boards, that can be used to integrate modules from different design teams or vendors into testable and maintainable subsystems. (In this document “board” is used in a broad sense to mean an assembled, usually field-replaceable, unit immediately above the device-level in a system’s assembly hierarchy—e.g., assembled printed wiring boards, assembled multichip modules, etc.) The bus protocol defined in this document standardizes a method for communication of test and maintenance commands and serial data between a subsystem test control module (MTM-Bus Master module) and the other modules (MTM-Bus Slave modules) on the bus. The MTM-Bus may be implemented as part of an overall test and maintenance interface architecture that includes other test buses.

1.1.2 Objectives

The MTM-Bus is intended for use in the testing, diagnosis, and maintenance of electronics subsystems and modules. Permissions and recommendations have been incorporated into the MTM-Bus protocol that allow it to be tailored for use in commercial computers, telecommunications, avionics, defense electronics, and many other applications. The MTM-Bus may be used to support the following:

- *Module test.* Testing of modules during manufacturing to provide fault isolation of defects to individual components.
- *Subsystem test.* Off-line testing of modules while contained within a subsystem, and the testing of interconnections between modules within a subsystem.
- *Subsystem diagnostics.* On-line diagnostics within a system to support logging of detected errors, initiation of self-test, reconfiguration of subsystem resources, and other diagnostic functions.
- *Software/hardware development.* Access to a module or subsystem state by use of low-level observability/controllability techniques (e.g., scan, boundary scan, etc.). Use of these techniques may allow for reduced hardware/software development costs and decreased time to market.

To support these applications, the MTM-Bus is designed as a serial backplane bus with a multidrop topology. As the multidrop bus signals are common between all modules, a board may be removed from the backplane without breaking the communication link between other modules in the backplane. With the appropriate Physical Layer design, the protocol supports access between a single MTM-Bus Master module and up to 250 individually addressable MTM-Bus Slave modules. Addressing is defined so that the MTM-Bus Master may communicate with one, some, or all of the Slaves at one time. The physical backplane characteristics will define the maximum number of MTM-Bus modules that can be used in a given application.

A primary concern in the development of the MTM-Bus Standard was to minimize the cost of implementation while still deriving the benefits of improved testability and maintainability. For this reason, a serial bus with a limited number of signals was selected. A master/slave protocol is used to keep the protocol simple; thereby reducing interface circuitry and software requirements. The bus is synchronous to simplify the backplane design and to simplify simulation of bus performance.

Because the MTM-Bus may be used to transmit a significant amount of data, such as serial test patterns, the protocol supports full duplex data transfer operations. Although the detailed timing and electrical parameters of the bus are not specified in this document, a general timing approach is defined that allows a MTM-Bus interface device to support a broad range of applications. The maximum bus performance for a particular system is governed by the maximum capabilities of the bus interface devices and the worst-case system timing budget. To aid interoperation of modules from multiple vendors, a dual-edge clocking approach is utilized. The dual-edge approach allows the system designer to select a MTM-Bus clock waveform that balances the system performance requirements with the modules' capabilities and the backplane timing parameters.

As the bus is intended to be usable in-system, the bus provides a number of fault detection and fault tolerance features. Optional features have been defined that may be included for a particular test architecture.

To support the requirements described previously, the protocol identifies commands to provide for the following functions:

- Initialization of module/subsystem
- Accessing lower-level test buses
- Testing of module interconnections
- Interrupt control and handling
- Accessing module identification
- Accessing module fault logs
- Forcing modules off- and on-line
- Accessing module built-in test features
- Private and public user extensions

1.2 The interconnection of modules with MTM-Bus

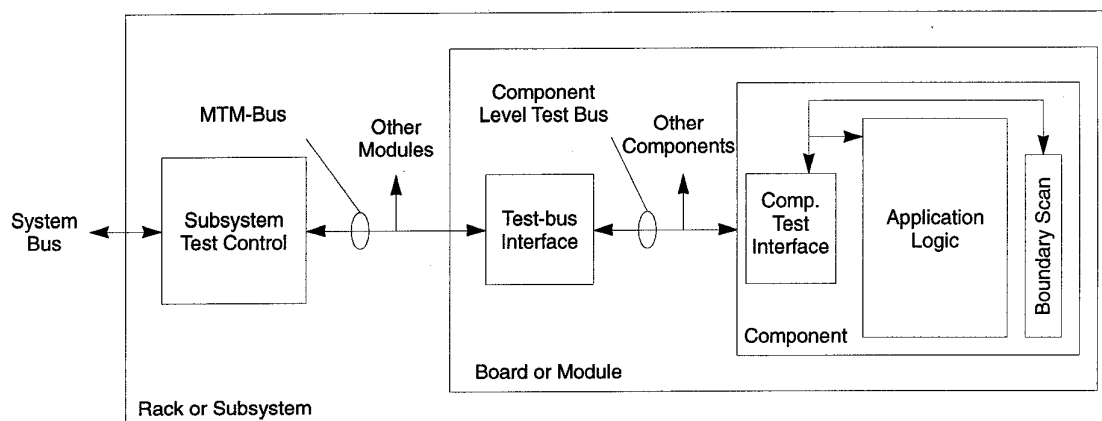
As designs have become more complex and difficult to test, design-for-test features (e.g., internal scan, IEEE Std 1149.1 boundary-scan, and built-in test) are being integrated into component designs to aid in component, board, subsystem, and system test and maintenance. Since these techniques do not use "functional" methods to test the circuitry, alternate paths are needed to access on-chip design-for-test features. These alternate paths have to be kept both simple and dedicated to test and diagnostic functions so that bootstrapping to complete module testing from a testable kernel is feasible.

When components are assembled on boards, boards into subsystems, and subsystems into systems, a hierarchy of test buses is needed to retain access to the design-for-test features in the final product. Such a hierarchy of test buses is depicted in figure 1-1. The MTM-Bus has been developed to meet the requirements for a backplane bus in such a bus hierarchy. As shown in figure 1-1, the MTM-Bus provides subsystem test control (e.g., maintenance module, service console, etc.) access or external test equipment access to test features on modules within a subsystem.

The actual use of the MTM-Bus will vary with the system test and maintenance strategy adopted. Two extremes would be highly "centralized" and highly "distributed" approaches. A distributed approach pushes as much of the test and maintenance function to the individual modules as possible. A centralized approach uses a centralized test control module or serial test equipment to sequence low-level module testing steps. The benefits of both distributed and centralized approaches are outlined in the following lists.

Benefits of distributed approach:

- *Interoperability.* In a system where modules are provided by a number of different design teams, a distributed approach allows backplane integration to proceed more smoothly.
- *Reduction in test time.* It is possible that test time at the system level can be reduced if two or more modules can be tested concurrently.



NOTE—Hierarchical serial test buses may be used to access module- and component-level design-for-test features.

Figure 1-1—Hierarchy of test and maintenance buses

- *Simplification of software.* Since the distributed test circuitry and software allows for higher-level test functions/commands, less software is required in the Subsystem Test Control module (figure 1-1) for each specific type of module. Distributed software can reduce the software development time and simplify module upgrades.
- *Containment of bus traffic.* If self-test routines and test patterns are contained on the individual modules, bus traffic is restricted to high-level commands and compressed test results.
- *Interchangeability.* If test patterns or procedures specific to a module's low-level design are contained within the module, then modules with identical function, but different design, can be made fully interchangeable.

Benefits of a centralized approach:

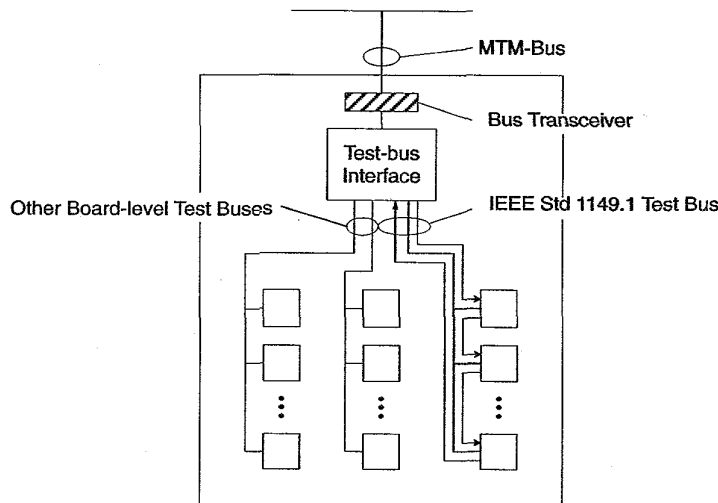
- *Cost reduction.* By centralizing the test sequencing function to a single processor on a Subsystem Test Control module (figure 1-1), test sequencing capability is not required on each module. This can reduce the cost of the test interface and control circuitry on each module within the system.
- *Simplicity.* If the test interface on each module does not contain any module-specific test information, a common test interface can be used on a large number of modules without need for module-specific programming.

1.3 Relationship to other test buses

The MTM-Bus is intended for use as a backplane serial test bus that may be used in parallel with, or in an hierarchy with, other test buses. The MTM-Bus and on-module test buses may be connected as shown in figure 1-2 to allow access to the component test features.

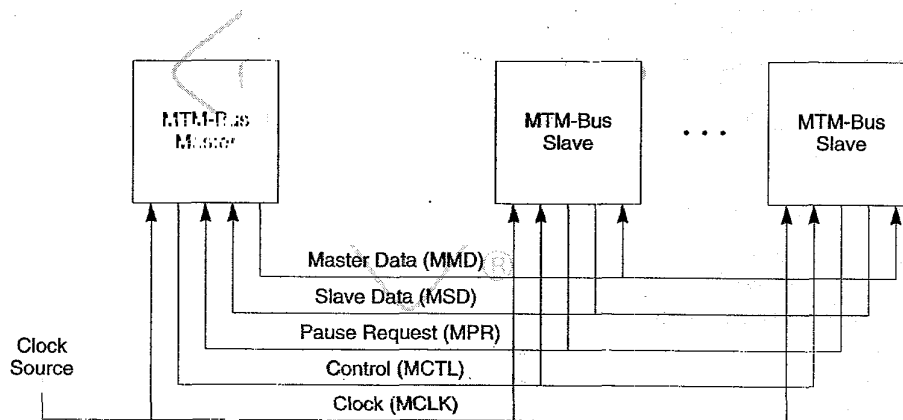
1.4 Overview of MTM-Bus operation/protocol

The MTM-Bus is a synchronous, serial, backplane bus comprised of four required signals and an optional fifth signal, as detailed in figure 1-3. An MTM-Bus Master module can communicate with up to 250 individually addressable MTM-Bus Slave modules. The bus is designed to have a single bus master; however, bus mastership may be transferred to backup masters in fault-tolerant configurations of the bus. The multidrop architecture and addressing capabilities of this Standard allow the MTM-Bus Master to address and communicate with one, some (multicast), or all (broadcast) of the MTM-Bus Slave modules on the bus.



NOTE—Translation may be provided by the use of a test-bus interface device. A separate bus transceiver may also be used to customize the physical protocol for specific applications.

Figure 1-2—Translation between the MTM-Bus and on-module test buses



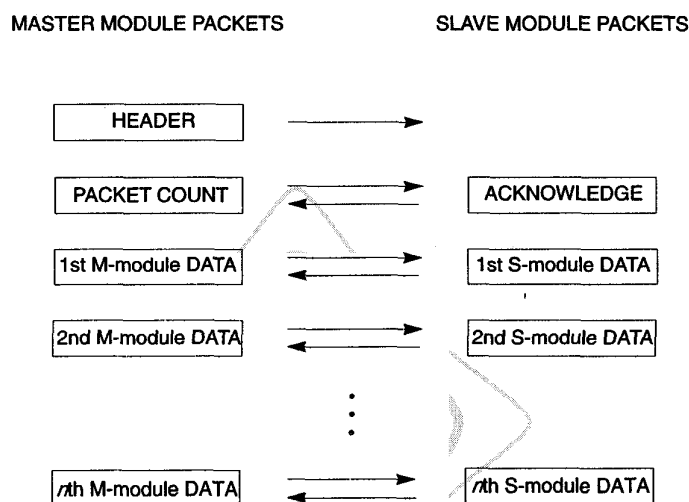
NOTE—These signals include the test-bus clock (MCLK [free-running and externally sourced in this example]), MTM-Bus Master and Slave Data signals (MMD and MSD), the Control signal (MCTL), and the recommended Pause Request signal (MPR).

Figure 1-3— Backplane MTM-Bus signals

The MTM-Bus Master communicates with MTM-Bus Slaves using messages that consist of a series of packet transfers. A message consists of a HEADER packet, an optional ACKNOWLEDGE/PACKET COUNT packet, and a variable number of DATA packets.

Figure 1-4 depicts a standard format for an MTM-Bus message. To start a message, the MTM-Bus Master transmits a HEADER packet that includes the addresses of the MTM-Bus Slaves who are to participate in the message sequence along with a command. If a single MTM-Bus Slave is addressed and the HEADER packet includes a request for an ACKNOWLEDGE packet, the MTM-Bus Master sends the PACKET

COUNT packet while the addressed MTM-Bus Slave is returning the ACKNOWLEDGE packet. This PACKET COUNT packet identifies how many packets the MTM-Bus Master expects to transfer during the message exclusive of HEADER and PACKET COUNT packets. The message may then include the transfer of DATA packets between the MTM-Bus Master and the single addressed MTM-Bus Slave module depending upon the command. If multiple MTM-Bus Slaves are addressed (i.e., via broadcast or multicast addressing), no ACKNOWLEDGE packet is sent and any transfer of DATA packets is only from the MTM-Bus Master to the MTM-Bus Slaves. In the broadcast and multicast cases, the MTM-Bus Master receive a constant logic 0—the value of the Slave Data line (figure 1-3) when it is not actively driven by any MTM-Bus Slave. The number of DATA packets to be transferred is dependent upon the type of command contained within the HEADER packet.



NOTE—MTM-Bus messages consist of a single HEADER packet, an optional ACKNOWLEDGE/PACKET COUNT packet pair, and a variable number of DATA packet pairs.

Figure 1-4—Sample MTM-Bus message format

All packet transfers occur under the control of the MTM-Bus Master module. Addressed MTM-Bus Slave modules may request that the MTM-Bus Master insert additional Pause states between data packet transfers by asserting the Pause Request (MPR) signal. This feature may be used to accommodate slow MTM-Bus Slaves or to simplify data flow control across asynchronous interfaces.

MTM-Bus Slave modules may also request the attention of the MTM-Bus Master by sending an interrupt, multiplexed on the Slave Data (MSD) signal wire, at any time between packet transfers. As there is only one MSD input to the MTM-Bus Master module, the MTM-Bus Master may need to identify the interrupting module via polling or by the Contend for Bus command (16.4.1).

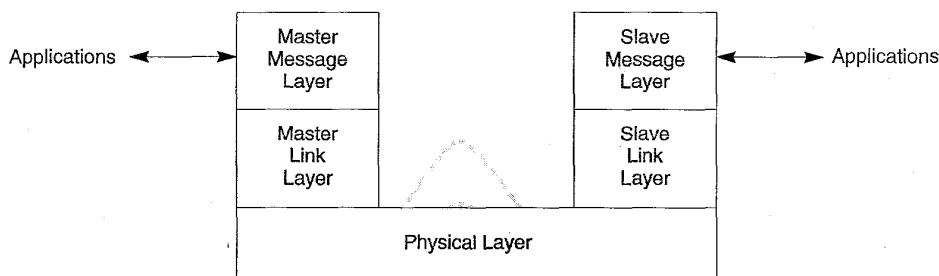
1.5 MTM-Bus protocol layers

The individual layers of the MTM-Bus protocol are identified in this document to simplify understanding of the bus protocol and to clarify bus requirements. The bus protocol includes Physical, Link, and Message Layers as shown in figure 1-5.

The Physical Layer requirements specify the physical interconnections that comprise the bus. This includes the specification of the minimum requirements regarding physical interconnections, signal characteristics, and timing characteristics. The minimum requirements for the Physical Layer are specified in clause 4. The Physical Layer characteristics that are specific to system technology are not defined in this Standard.

The architecture of the MTM-Bus and its addressing scheme are specified in clause 3.

The Link Layer requirements specify the protocol features that permit error-free packet transfer between the MTM-Bus Master and one or more MTM-Bus Slaves. These features include serialization/packetization of information, parity generation and checking, and address recognition. The Link Layer also provides for the multiplexing of data and control functions on single wires. The requirements for the Link Layer are specified in clauses 5 through 11 of this Standard.



NOTE—The MTM-Bus protocol includes Physical Layer, Link Layer, and Message Layer protocols.

Figure 1-5—Protocol layers

Message layer requirements specify the syntax for messages and identifies the functions that have to or can be supported by the MTM-Bus Master or MTM-Bus Slaves. Requirements for the Message Layer are specified in clauses 12 through 20.

Port requirements describe a set of recommended or permitted data sources/destinations (called ports) that may be useful in some S-modules. A port may be as simple as a register or as complex as an application interfacing this Standard to an on-module test bus. Data is transferred to/from ports during execution of Data Transfer class (15.1.1; clause 17) commands. Requirements for ports are found in clause 21. Requirements for recommended ports interfacing an MTM-Bus to the IEEE Std 1149.1 test bus are found in clauses 21.10 through 21.12.

Documentation requirements are to be found in clause 22.

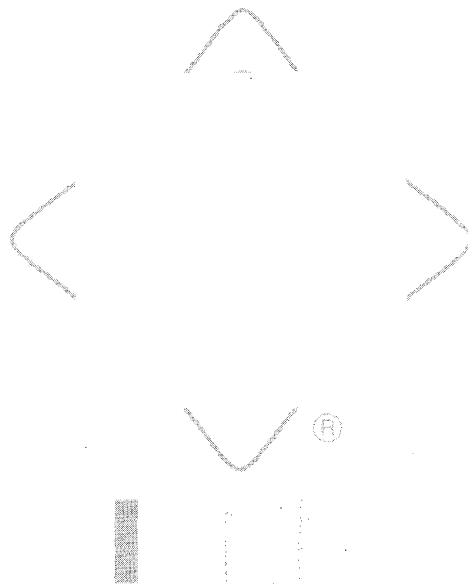
1.6 Extensions to this Standard

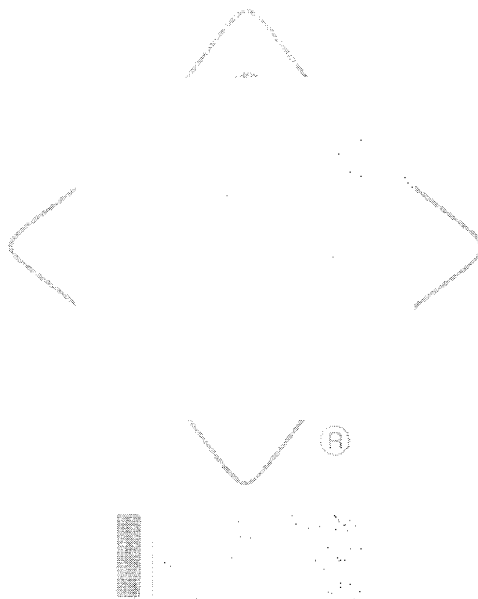
The basic protocol defined herein for the MTM-Bus may be extended to meet the needs of specific applications—as detailed by the recommendations and permissions. These extensions may address such areas as fault tolerance, electrical characteristics, and others that allow the MTM-Bus to be adapted to user applications.

An example of one of these MTM-Bus-based extensions is a test and maintenance bus system for avionics applications that is being developed by the Society of Automotive Engineers (SAE) Avionics Systems Division members (see SAE AS4765).¹ This system will contain dual MTM-Buses (2 sets of 5-wire MTM-

¹Information on references can be found in 2.4.

Buses), a fault recovery protocol, and additional Physical Layer requirements. This system will also include a protocol allowing "backup" MTM-Bus Master modules to monitor bus traffic to verify that the current MTM-Bus Master module is functioning correctly.





2. Conventions, definitions, and references

2.1 Document outline

Bus protocols such as that defined by this Standard are more easily understood if their specifications are accompanied by general descriptive material that places the details of the various parts of the protocol in perspective and provides implementation examples. Clause 1 contains an overview of the application of this Standard to test and maintenance of electronics subsystems. Clause 2 contains reference information that should be helpful in interpreting the subsequent clauses of the Standard. Clauses 3 through 22 contain the specifications for particular features of this Standard. The two classes of material contained in clauses 3 through 22 are described in 2.1.1 and 2.1.2.

2.1.1 Specifications

Subclauses entitled “Specifications” contain the rules, recommendations, and permissions that define this Standard:

Rules specify the mandatory aspects of this Standard. Specifications that are rules contain the word *shall*.

Recommendations indicate preferred practice for designs that seek to conform to this Standard. Specifications that are recommendations contain the word *should*.

Permissions show how optional features may be introduced into a design that seeks to conform to this Standard. These features will extend the application of the protocol defined by the Standard. Specifications that are permissions contain the word *may*.

2.1.2 Descriptions

Material not contained in subclauses entitled “Specifications” is descriptive material that illustrates the need for the features being specified, an example of their application, or other relevant information. This material includes bus packet or message sequences or schematics that illustrate a possible implementation of the specifications in this Standard. The descriptive material also discusses design decisions made during the development of this Standard.

NOTE—The descriptive material contained in this Standard is for illustrative purposes only and does not define a preferred implementation.

2.2 Conventions

The following conventions are used in this Standard:

- a) The rules, recommendations, and permissions in each Specifications subclause are contained in a single alphabetically indexed list. References to each rule, recommendation, or permission are written in the following form:

15.1.1c(ii)
 └──────────┘
 Subclause number
 Index (if any)

- b) State and packet names defined in this Standard are written entirely in capital letters in the text of this Standard.

- c) A positive logic convention is used for all figures in this document, i.e., a logic 1 value designates the most positive of the two voltages used for signals, including the MTM-Bus clock signal, MCLK.
- d) When a set of related bits are denoted by a common name, the bits within the set are identified by number with the least significant bit numbered 0.
- e) The identifier for a single bit in a series of related bits (e.g., a packet) is the bit position number in the series enclosed in angle brackets (“<” and “>”).
- f) The expression <*m*..*n*> is used as an abbreviated identifier for a series of bits *m* to *n*, inclusive (e.g., the bits in a register or packet), where *m* and *n* are the most and least significant bits, respectively.
- g) The phrases “signal driver(s),” “signal output(s),” and “signal input(s),” when implicitly or explicitly referring to MTM-Bus signals, shall be taken to refer to one or more MTM-Bus signals *exclusive* of MCLK unless explicitly indicated otherwise.
- h) *M-module* is used as an abbreviation for “MTM-Bus Master module.”
- i) *S-module* is used as an abbreviation for “MTM-Bus Slave module.”
- j) In all diagrams depicting registers, packets, and data fields within registers and packets, the most significant bit (msb) of each depicted element is assumed to be left-most within the rectangle defining the data entity.
- k) Any reference to a figure in this document is intended to reference all drawings and text in that figure.

2.3 Definitions

The following terms are used within this Standard. Clause, subclause, table, or figure numbers are given in parentheses to indicate specific places in the text where terms are discussed in detail.

2.3.1 ACKNOWLEDGE packet: The first packet returned by an individually addressed S-module that conveys to the M-module that the appropriate S-module is responding and indicates the current status of the responding S-module (5.3).

2.3.2 active: When associated with a logic level (e.g., in the word “active-low”), this term identifies the logic level to which a signal shall be set to cause a defined action to occur. When referring to an output driver (e.g., in the phrase “an active driver”), this term describes the mode in which the driver is capable of determining the voltage of the network to which it is connected. 

2.3.3 amnesia address: The module address (‘FA’ HEX) to which a module will respond as though uniquely addressed if that module (1) implements the ability to detect when it cannot determine its address unambiguously and (2) detects that it cannot determine its address unambiguously (3.3).

2.3.4 application logic: That portion of a module that excludes the MTM-Bus interface logic (figure 2-1).

2.3.5 assert: To change the value of a bus signal from logic 0 (released) to logic 1 (asserted) or ensure that such a signal remains at a logic 1.

2.3.6 asserted: Having a current value equal to logic 1 (said of any signal).

2.3.7 backplane: Motherboard comprising connectors for the modules of a system and wiring interconnecting those modules. The intermodule wiring of the MTM-Bus is expected to be on this motherboard.

2.3.8 BMR: Acronym for “broadcast/multicast received.” See 2.3.10.

2.3.9 broadcast: A mode of operation of the MTM-Bus in which an MTM-Bus Master transmits data to all connected S-modules simultaneously throughout a message. Also, a message transmitted in this mode.

2.3.10 Broadcast/Multicast Received (BMR) bit: A bit in the Bus Error register of all S-modules. An S-module sets this bit to indicate that the last broadcast or multicast message was received without error (9.3).

2.3.11 BSE: Acronym for "Bus Error." (See 2.3.14.)

2.3.12 BSY: Acronym for "Slave Busy." (See 2.3.123.)

2.3.13 BTL: Acronym for "backplane transceiver logic."

2.3.14 Bus Error (BSE) bit: A bit in the Slave Status register of every S-module that is set by the S-module when a Bus Error is recorded in the Bus Error register (9.2).

2.3.15 Bus Error register: A status register that is required to be implemented in the MTM-Bus interface circuitry of every S-module. Bits in this register provide the S-module with the ability to record error conditions associated with message transmission. The register may be interrogated by the M-module. Some bits in the register are reserved for application-specific uses (9.3).

2.3.16 class: *See: commands, class of; messages, class of.*

2.3.17 clear: To force the contents of one or more storage elements to the logic 0 state.

2.3.18 clock: A signal, the transitions of which (between the low and high logic level [or vice versa]) are used to indicate when a stored-state device, such as a flip-flop or latch, may perform an operation.

2.3.19 collision: The condition occurring when two MTM-Bus modules are simultaneously MTM-Bus drivers and are attempting to drive a MTM-Bus signal to complementary values (one driving logic 1, one driving logic 0).

2.3.20 commands, class of: One of the groups of MTM-Bus commands. Every MTM-Bus command is assigned to a command class (15.1).

2.3.21 Command Resource Unavailable (CRU) bit: A bit in the Bus Error register of all S-modules. An S-module sets this bit to indicate that resources required to complete execution of a command were not available and that the command was not executed.

2.3.22 Command Sequence Error (CSE) bit: A bit in the Bus Error register of all S-modules. An S-module sets this bit to indicate that the module has received a command that requires a previous enabling command without receipt of such an enabling command (9.3).

2.3.23 command X, receipt of: Error-free receipt of the HEADER packet containing in its Command field the command code of X.

2.3.24 contend: To actively and simultaneously vie for the attention of the MTM-Bus Master module (said of a group or one or more S-modules).

2.3.25 critical control command: An MTM-Bus command that has significant effect on the operation of a module to a degree that, for added security, a message conveying such a command should be difficult to send unintentionally. This Standard provides that a message containing a critical control command has to be preceded by an Enable Module Control (EMC) message (15.1; table 15.1; 16.10). If this procedure is not followed, a Command Sequence Error (11.9) will occur.

2.3.26 CRU: Acronym for "Command Resource Unavailable." (See 2.3.21.)

2.3.27 CSE bit: Acronym for "Command Sequence Error." (See 2.3.22.)

2.3.28 Data Overrun Error (DOR) bit: A bit in the Bus Error register of all S-modules. An S-module sets this bit to indicate that the module has received input data from the M-module when the S-module was not ready to receive it (9.3).

2.3.29 DATA packet: Any packet other than a HEADER, PACKET COUNT, or ACKNOWLEDGE packet.

2.3.30 DOR: Acronym for "Data Overrun Error." (See 2.3.27.)

2.3.31 ECL: Acronym for "emitter coupled logic."

2.3.32 EMC: Acronym for "Enable Module Control."

2.3.33 error bit: A bit in a status register of an S-module that is associated with detection of some error detected by that S-module (11.1.1). Such bits may be found in the Bus Error register (9.3.1), the optional Module Status register (9.4.1), or in an Additional Status register (9.5.1). Error bits of the Bus Error register affect the value of the BSE bit of the Slave Status register (9.2.1). Error bits of the optional Module Status register or of an Additional Status register are permitted to affect the value of either the BSE bit or EVO bit of the Slave Status register.

2.3.34 Event Occurrence (EVO) bit: A bit in the Slave Status register of every S-module that is set by the S-module when a module-application-related condition requiring an interrupt has occurred (9.2).

2.3.35 EVO bit: Acronym for "Event Occurrence." (See 2.3.34.)

2.3.36 falling edge: The transition from a logic one to logic zero.

2.3.37 fsm: Acronym for "finite state machine."

2.3.38 HEADER packet: A packet originating in the MTM-Bus Master that is the first packet of an MTM-Bus message. The HEADER packet includes an address and a command field. The address identifies which S-module(s) are to interpret and act upon the command contained within the command field (5.2).

2.3.39 IBIT: Acronym for "initiated built-in test."

2.3.40 Idle Interrupts Enabled (IIE) bit: A bit in the Slave Status register of every S-module that is set to indicate that the S-module may generate an interrupt during S-idle states (9.2).

2.3.41 idle state: Any Link Layer Controller state the name of which begins with the uppercase letters "IDLE" (6.1.1; 7.1.1). Such states in the MTM-Bus Master Link Layer Controller are called M-idle states and in the MTM-Bus Slave Link Layer Controller are called S-idle states.

2.3.42 IIE: Acronym for "Idle Interrupts Enabled." (See 2.3.40.)

2.3.43 ILC: Acronym for "Illegal Command." (See 2.3.44.)

2.3.44 Illegal Command (ILC) bit: A bit in the Bus Error register of all S-modules. An S-module sets this bit to indicate that the module has received an illegal command (9.3; 16.17).

2.3.45 inactive: When referring to an output driver (e.g., in the phrase "an inactive driver"), this term describes the mode in which the driver is *not* capable of determining the voltage of the network to which it is connected.

2.3.46 Incorrect Packet Count (IPC) bit: A bit in the Bus Error register of all S-modules. An S-module sets this bit to indicate that, with respect to a just completed message transfer, either the S-module has received a request for an ACKNOWLEDGE packet and was not given the opportunity to send it or, in the case of an S-module in which the packet counting option is implemented (14.2.1), that it received a different number of packets than was specified in the PACKET COUNT packet (9.3).

2.3.47 interoperable: Said of two modules indicating that they may both be placed on the same physical MTM-Bus without causing errors of operation.

2.3.48 interchangeable: Said of two modules that, although possibly of different design, perform identical functions and have identical interface characteristics.

2.3.49 Illegal Port Selected (IPS) bit: A bit in the Bus Error register of all S-modules. An S-module sets this bit to indicate that the module has received a command addressed to an unsupported port (9.3).

2.3.50 IPC: Acronym for "Incorrect Packet Count." (See 2.3.46.)

2.3.51 IPS: Acronym for "Illegal Port Selected." (See 2.3.49.)

2.3.52 least significant bit (lsb): The bit having the smallest effect on the value of a binary numeral; usually the right-most bit in figures.

2.3.53 least significant word (lsw): In a multiword representation of a binary number, the word containing the lsb of that number.

2.3.54 logic 0 (logic zero): The lowest voltage value of the two logic levels on an active-high signal and the highest voltage value of the two logic levels on an active-low signal.

2.3.55 logic 1 (logic one): The highest voltage value of the two logic levels on an active-high signal and the lowest voltage value of the two logic levels on an active-low signal.

2.3.56 lsb: Acronym for "least significant bit." (See 2.3.52.)

2.3.57 lsw: Acronym for "least significant word." (See 2.3.53.)

2.3.58 Master-capable: Said of an MTM-Bus module that is an S-module at a given time, but contains appropriate circuitry so that it may be converted by system control to an M-module if required.

2.3.59 Master Controller state: A state of the fsm (figure 6-1) required of M-modules that controls M-module Link Layer behavior with regard to message transmission.

2.3.60 M-Controller: Abbreviation for "MTM-Bus Master Link Layer Controller."

2.3.61 MFS bit: Acronym for "Module Fail Status." (See 2.3.72.)

2.3.62 message: A set of packets starting with a HEADER, consisting of that HEADER and all (ACKNOWLEDGE and DATA) packets transmitted as the immediate consequence of the command in that HEADER, and terminating when the M-module returns to the IDLE Master Controller state.

2.3.63 messages, class of: A group of messages having in the Command fields of their respective HEADER packets command codes of commands belonging to a single class of commands (table 15-1). The name, C, of a class of messages is the same as the name of the class of commands that defines the class C.

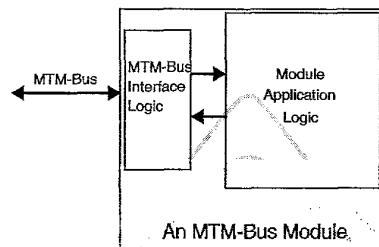
2.3.64 messages, species of: A group of messages having in the Command fields of their respective HEADER packets a common command code (table 15-1). The name, *S*, of a message species is the same as the name of the command that defines the message species.

2.3.65 MICT: Acronym for “Module I/O Control and Test” (clause 19).

2.3.66 MIST: Acronym for “Module Initialization and Self-Test” (clause 18).

2.3.67 M-module: Abbreviation for MTM-Bus Master module.

2.3.68 module: An addressable unit or interconnected set of units attached to the MTM-Bus and fully supporting the MTM-Bus protocols. The boundary of an MTM-Bus module may correspond to the physical partitioning of the system, but is not required to do so. For the purposes of this document, a module is comprised of an MTM-Bus interface and module application logic, as shown in figure 2-1.



NOTE—An MTM-Bus module consists of MTM-Bus interface logic and module application logic.

Figure 2-1—MTM-Bus module

2.3.69 module address: An eight-bit value uniquely identifying an MTM-Bus module.

2.3.70 Module Fail Status (MFS) bit: A bit in the Slave Status register of every S-module that is set by the S-module when the module’s built-in test has failed or is currently executing (9.2).

2.3.71 Module Status register: A status register that is recommended to be implemented in the MTM-Bus interface circuitry of every S-module. The bits in this register are defined by the manufacturer of the MTM-Bus interface circuitry of an S-module. The bits of such a register may serve to record error-condition detection or the module’s application-related status (9.4).

2.3.72 most significant bit (msb): The bit having the greatest effect on the value of a binary numeral; usually the left-most bit in figures.

2.3.73 most significant word (msw): In a multiword representation of a binary number, the word containing the msb of that number.

2.3.74 msb: Acronym for “most significant bit.”

2.3.75 MSB0; MSB1: Acronyms for “multicast select bit 0” and “multicast select bit 1.”

2.3.76 MSD: Acronym for “MTM-Bus Slave Data.”

2.3.77 msw: Acronym for “most significant word.” (See 2.3.76.)

2.3.78 MTM-Bus: A serial, backplane, test and maintenance bus, consisting of one or more logic boards, that can be used to integrate modules from different design teams or vendors into testable and maintainable subsystems, as specified in this Standard.

2.3.79 MTM-Bus interface logic: The portion of a module that is designed for the purpose of MTM-Bus-compliant communication and through which takes place all the communication between the given module and any other on a given MTM-Bus implementation (7.2). MTM-Bus interface logic need not be defined on the basis of physical package boundaries.

2.3.80 MTM-Bus Master: The module in control of the MTM-Bus. This is the module that, at a given time, is sourcing MCTL and MMD.

2.3.81 MTM-Bus mastership: Property of being the current MTM-Bus Master module.

2.3.82 MTM-Bus Slave: An MTM-Bus module that cannot command actions of other modules on the bus, but that may be selected by the MTM-Bus Master module to participate in a message.

2.3.83 multicast: A mode of operation in which the M-module transmits data simultaneously (i.e., during a single message) to a predefined subset of the S-modules currently connected to the bus. Also, a message transmitted in this mode.

2.3.84 multicast select bit 0; multicast select bit 1 (MSB0; MSB1): Those bits in the Slave Status register of every S-module by means of which the S-module is programmed to be a member of one of the four possible multicast select groups.

2.3.85 multicast select group: A group of S-modules that may be addressed simultaneously in a multicast. Four such groups are possible. Each has an address defined by this Standard (3.3). The multicast select group of an S-module is programmable. *See also:* **multicast select bit 0; multicast select bit 1 (MSB0; MSB1).**

2.3.86 multidrop: Said of the configuration of a bus with a single shared medium segment that allows one or more of its module connectors to be unoccupied without disturbing bus operation.

2.3.87 non-NULL DATA packet: A DATA packet other than a NULL packet.

2.3.88 null DATA packet: *See:* NULL packet.

2.3.89 NULL packet: A special type of DATA packet containing a data field entirely filled with logic zeros and a parity bit equal to logic one (5.6). *Syn:* null DATA packet.

2.3.90 off-line: Used to describe an MTM-Bus module when it is in a mode such that it will not respond to state transitions on MTM-Bus signals whether or not the module is connected to the bus. Also used to describe such a mode.

2.3.91 packet: A 17-bit unit of data consisting of a 16-bit word plus 1 parity bit.

2.3.92 PACKET COUNT packet: A packet by means of which an M-module conveys to addressed S-modules the number of packets to follow in the current message. S-modules may or may not include the ability to use the data in this packet (5.4).

2.3.93 packet latency: During a send/receive message, the number of packet transfers by which the first non-NULL DATA packet returned by the addressed S-module lags the first non-null DATA packet transmitted by the M-module.

2.3.94 packet pair: Two packets, one transmitted by the M-module and one by an S-module, such that the last 14 bits of the M-module-originated packet are transmitted simultaneously with the first 14 bits of the S-module-originated packet.

2.3.95 Parity Error (PRE) bit: A bit in the Bus Error register of all S-modules. An S-module sets this bit to indicate that the module has detected a Parity Error on a DATA packet received (9.3).

2.3.96 participate: With regard to the action of an S-module during message transmission, to execute the command contained in the HEADER packet of the current message and return packets as required by that command and by the state of the Acknowledge bit in the HEADER packet. The handling of some errors may cause an S-module to cease to participate in a message (clause 11)—e.g., by ceasing to execute the current command, by returning NULL packets when data is expected, by driving a constant value on the MSD signal without regard to packet transmission timing, etc.

2.3.97 Pause Interrupt Enabled (PIE) bit: A bit in the Slave Status register of every S-module that is set to indicate that the S-module may generate an interrupt during the PAUSE Slave Controller state when the S-module is addressed (9.2).

2.3.98 PIE: Acronym for “Pause Interrupt Enabled.” (See 2.3.101.)

2.3.99 port: A source or destination of data transferred by a Data Transfer class command into and/or out of an S-module (clause 21). A port may be an on-module memory, on-module interface, a peripheral attached to a module, or some other mechanism to/from which data is passed. Within this Standard, a port is defined by a module address, a port ID meaningful to the MTM-Bus interface logic of that module, and the semantics and structure of packets by which data can be conveyed to and/or from that port. This latter often entails some description of the application to/from which data are passed. A port is selected/accessed/addressed via a Data Transfer class command.

2.3.100 PORT ID packet: The first DATA packet transferred in a Data Transfer class message. This packet contains the identifier (Port Identifier) by which means a port is selected for the remainder of the message.

2.3.101 Port Transfer Error: A port-specific error indicating some failure with relation to transmission of command or data to/from a currently selected port.

2.3.102 Port Transfer Error (PTE) bit: A bit in the Bus Error register of all S-modules. An S-module sets this bit to indicate that the port selected by the command in the current message has reported a Data Transfer Port Error. Such errors will be found defined in the port documentation of specific S-modules. The acronym stands for “Port Transfer Error” (9.3).

2.3.103 PRE: Acronym for “Parity Error.” (See 2.3.99.)

2.3.104 PTE: Acronym for “Port Transfer Error.” (See 2.3.106.)

2.3.105 release: To cease to assert a logic 1 on a bus signal line. (One module’s releasing a signal line produces a change in value of the signal line only if no module is asserting the signal.)

2.3.106 released: Having a value equal to logic 0 (said of any signal). Equivalently, in the case of an MTM-Bus signal line, not asserted by any module on the bus.

2.3.107 reset: When describing the operating status of an S-module, the state of the S-module’s Status registers defined by 9.2.1b, 9.3.1b, 9.4.1b, and 9.5.1b—the state of these registers produced by execution of the Reset Slave Status command (16.3.1).

2.3.108 response: In the context of message transmission, the set of packets sent by an S-module during a single message. In the context of the operation of S-modules, an S-module's action that is a direct consequence of the command most recently received by that S-module.

2.3.109 rising edge: A transition from a logic zero to a logic one. (See t_{LH} in 4.4 and in figure 4-4.)

2.3.110 S-Controller: Abbreviation for "MTM-Bus Slave Link Layer Controller."

2.3.111 S-module: Abbreviation for "MTM-Bus Slave module."

2.3.112 SBIT: Acronym for "Start-up Built-in Test" (performed at power-up or after resets of a module).

2.3.113 SDF: Acronym for "Slave Data Fault." (See 2.3.125.)

2.3.114 select: To identify and fix (for the duration of the current message) a port to which data of a Data Transfer class message are to be directed.

2.3.115 sequence: A set of bits, packets, or messages ordered in time and that are, or that are intended to be, transmitted consecutively without interruption.

2.3.116 set (used as a verb): To force the contents of one or more storage elements to a logic 1.

2.3.117 simple message: An MTM-Bus message that consists of no more than a HEADER and, optionally, an ACKNOWLEDGE/PACKET COUNT packet pair.

2.3.118 singlecast: A mode of operation in which the M-module transmits data to a single S-module. Also, a message transmitted in this mode.

2.3.119 Slave Busy (BSY) bit: A bit in the Slave Status register of every S-module that is set by the S-module when the application logic of the S-module is executing a previously transmitted MTM-bus instruction or is executing its power-up sequence.

2.3.120 Slave Controller state: A state of the fsm (7.1.1) required of S-modules that controls S-module Link Layer behavior during message transmission to the module.

2.3.121 Slave Data Fault (SDF) bit: A bit in the Bus Error register of all S-modules. An S-module sets this bit when it is transmitting on the MSD signal and detects a fault on that signal.

2.3.122 Slave Status register: A status register that is required to be implemented in the MTM-Bus interface logic of every S-module. Bits in this register are used to record such items of module status as interrupt enable status, whether an error condition has occurred, whether a module application-related error has occurred, whether the module has failed its Built-In Self Test, etc.

2.3.123 SSE: Acronym for "State Sequence Error." (2.3.128.)

2.3.124 State Sequence Error (SSE) bit: A bit in the Bus Error register of all S-modules. An addressed S-module sets this bit to indicate that the S-module's Slave Link Layer Controller has entered the ERROR Slave Controller State (9.3).

2.3.125 status bit: A bit used to indicate a non-error condition important to S-module operation. Status bits are located in an S-module's Slave Status register (9.2.1) and Bus Error register (the BMR bit) (9.3.1) and may be located in the optional Module Status register (9.4.1) or an Additional Status register (9.5.1). Status bits in the Module Status register or in an Additional Status register are permitted to affect the value of the EVO bit of the Slave Status register.

2.3.126 status register: A register in an S-module by means of which current operating conditions of the S-module (e.g., interrupt enabled, module pass/fail status, multicast address of the S-module, etc.) and event occurrence (e.g., detection of an error condition during transmission of a message) can be recorded either for later interrogation by the M-module or to record the necessity of particular S-module activity at a later time.

2.3.127 transfer: The successful movement of a bit or bits between an MTM-Bus Master module and one or more modules co-connected by the MTM-Bus.

2.3.128 transfer state: A state of a Link Layer Controller the name of which begins with the letters "XFER" (6.1.1; 7.1.1). Such states in the MTM-Bus Master Link Layer Controller are called *M-transfer states* and in the MTM-Bus Slave Link Layer Controller are called *S-transfer states*.

2.3.129 TTL: Acronym for "transistor-transistor logic."

2.3.130 uniquely addressed: Said of an MTM-Bus S-module participating in a singlecast.

2.3.131 word: An ordered set of 16 bits operated on as a unit. The most significant bit is labeled bit 15 and the least significant bit is labeled bit 0.

NOTE—When a word of data is embedded in a MTM-Bus packet, bit <0> of the data word is placed in bit <1> of the 17-bit packet. See clause 5.

2.4 References

The following publications shall be used in conjunction with this Standard. When standards are referred to in this document, the latest revision shall apply:

IEEE Std 100-1992, The New IEEE Standard Dictionary of Electrical and Electronic Terms.²

IEEE Std 802-1990, IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture.

IEEE Std 1149.1-1990, IEEE Standard Test Access Port and Boundary-Scan Architecture (Includes IEEE Std 1149.1a-1993).

IEEE Std 1149.1b-1994, Supplement to IEEE Std 1149.1-1990, IEEE Standard Test Access Port and Boundary-Scan Architecture.

SAE Avionics Draft Standard AS4765, SAE Avionics TM-Bus Interoperability Standard, Revision 1.2, Nov. 1994.³

²IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.

³SAE publications are available from the Society of Automotive Engineers, 400 Commonwealth Drive, Warrendale, PA 15096, USA.

3. MTM-Bus architecture

This clause defines the component signals of the MTM-Bus, the interconnection of modules on the bus, the limitation of the number of such connections, and the conventions for addressing S-modules connected to the MTM-Bus. The MTM-Bus is described as a backplane bus; however, it may be used with cable or other connections that allow compliance with the MTM-Bus Physical Layer rules in clause 4.

3.1 MTM-Bus signals and interconnection of MTM-Bus modules via these signals

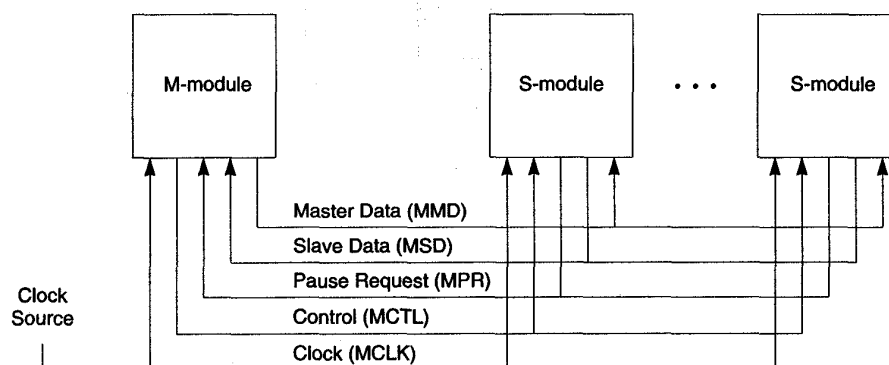
3.1.1 Specifications

Rules

- (a) The MTM-Bus shall include an MTM-Bus Control signal, MCTL, that is unidirectional from the current M-module to all connected S-modules.

NOTE—Since some S-modules may be *master capable*, we have to distinguish the current M-module as the one to which these specifications apply. In later clauses, we will simply say *the M-module* to refer to the current M-module.

- (b) The MTM-Bus shall include an MTM-Bus Master Data signal, MMD, that is unidirectional from the current M-module to all connected S-modules.
- (c) The MTM-Bus shall include an MTM-Bus Slave Data signal, MSD, that is unidirectional from each connected S-module to the current M-module.
- (d) The MTM-Bus shall include an MTM-Bus Clock signal, MCLK, that is unidirectional from the bus clock source to the M-module and connected S-modules.
- (e) All MTM-Bus signal inputs and outputs on a MTM-Bus module shall be dedicated connections (i.e., such module pins shall be used for no other purpose).
- (f) The MMD, MSD, MPR, and MCTL signals shall be connected to all modules in a multidrop configuration (i.e., all drivers/receivers for each of the four signals shall share a single physical connection as in figure 3-1).



NOTE—The MTM-Bus includes the Master Data (MMD), Slave Data (MSD), Control (MCTL), Clock (MCLK [free-running and externally sourced in this example]), and recommended Pause Request (MPR) signals.

Figure 3-1—Backplane MTM-Bus signals

Recommendations

- (g) The MTM-Bus should include an MTM-Bus Pause Request signal, MPR, that is unidirectional from connected S-modules to the current M-module.

3.1.2 Description

Dedicated module connections for MTM-Bus signals (3.1.1e) are required to allow access to the full range of mandatory features of this Standard. The five signals that make up the MTM-Bus are shown in figure 3-1. The MPR signal is a recommended signal that can be used to simplify the use of the bus in a system composed of modules having a wide range of operational speeds

MCTL—The M-module's link-layer protocol uses the MTM-Bus Control signal to force data transfer operations over the MTM-Bus MMD and MSD signals. When the MCTL signal is asserted, either a data transfer is occurring or there is an error condition. The MCTL signal is released during a pause in the transmission of a message, during the idle period between the transmission of messages, or in error conditions.

MMD—The MTM-Bus MMD signal is used to serially transfer control or data from the M-module to S-modules. Whether the value of the MMD signal is intended to convey control or data is dependent upon the M-module's Master Controller state.

MSD—The MTM-Bus MSD signal is used to transfer serial data from S-modules to the M-module using a logical-OR connection. The MSD signal is used for S-module data transfer and may be used to indicate an interrupt during pauses in message transmission and or in the idle period between the transmission of messages.

MCLK—The MTM-Bus MCLK signal is used to synchronize the transfer of data between modules on the MTM-Bus. All other MTM-Bus signals change value only at the rising edge of the MCLK signal, and MTM-bus modules capture these other signal values only at the falling edge of the MCLK signal.

MPR—The MTM-Bus MPR signal is used so that addressed S-modules may request the M-module to expand the pause between packet transfers during transmission of a message. This mechanism can be used to eliminate errors caused by the M-module sending packets too fast for a receiving S-module or by an S-module's not being ready to return data. Without the capability offered by the MPR signal, occurrence of such an error would cause an S-module to generate an interrupt to the M-module. This, in turn, could cause the M-module to abort the current message and investigate the cause of the interrupt. In some cases, this might lead to extremely lengthy message transfer times, even for very short messages.

3.2 General architecture and MTM-Bus mastership

3.2.1 Specifications

Rules

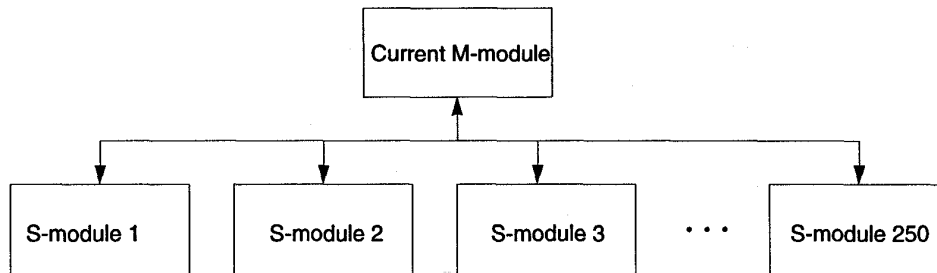
- (a) At any time that any module among the set of modules connected to the MTM-bus is operating as an M-module, exactly one of them shall operate as an M-module.
- (b) The MTM-Bus shall provide logical addressing for up to 250 modules.

Permissions

- (c) To improve bus availability and fault tolerance, MTM-Bus mastership may be transferred to a Master-capable S-module.

3.2.2 Description

The basic bus topology is shown in figure 3-2. In this diagram, a single M-module is depicted with a multi-drop connection to as many as 250 S-modules. The MTM-Bus modules need not necessarily map one-to-one to the physical partitioning of the system. It is allowable to define the system topology such that a single “module” actually consists of multiple logic boards or that multiple “modules” may reside on a single logic board.



NOTE—MTM-Bus topology allows for a single M-module and up to 250 S-modules.

Figure 3-2—MTM-Bus topology

Should system diagnostic routines detect a malfunction in the M-module of a particular implementation of the MTM-Bus, system software may reconfigure the system by deactivating the current M-module and making a Master-capable S-module the new M-module. If a system’s S-modules are designed in such a way as to make this possible, then much of the testing function can be preserved in the face of M-module failure or degradation.

3.3 S-Module addressing requirements

3.3.1 Specifications

Rules

- (a) In a given MTM-Bus implementation at a given time, any S-modules
 - (i) shall have at least one module address that is an eight-bit binary number between ‘0’ and ‘F9’ HEX;
 - (ii) shall include in its manufacturer’s documentation a description of the mechanism by which that address is, or shall be, defined.
- (b) In a given MTM-Bus implementation at a given time, no two S-modules connected to that MTM-Bus may have the same module address.

NOTE—This rule implies that generic or off-the-shelf S-modules have to have one or more module addresses that can be fixed by an user during assembly of a system (e.g., by the operation of switches or by programming).

- (c) An S-module’s module address shall be accessible to (i.e., readable by) the MTM-Bus interface circuitry of that module.

- (d) An S-module shall be considered *addressed* only in one of the following conditions:
- (i) The given S-module is able to determine its module address unambiguously; *and* the HEADER packet (5.2) received *without detection of an error* at the start of the current message contains either the S-module's module address, the broadcast address ('FB' HEX), or the multicast address ('FC', 'FD', 'FE', or 'FF' HEX) corresponding to the module's current multicast group (9.2; 16.5.1a).
 - (ii) The given S-module is unable to determine its module address unambiguously; *and* the HEADER packet received *without detection of an error* at the start of the current message contains either the amnesia address ('FA' HEX), the broadcast address ('FB' HEX), or commands addressed to the multicast address ('FC', 'FD', 'FE', or 'FF' HEX) corresponding to the S-module's current multicast group.

NOTE—It may be unwise to broadcast messages when more than one S-module in a system are responding to the amnesia address. No S-module can become addressed as a result of receiving an HEADER packet if an error such as bad parity in that HEADER packet (5.2.1; 11.5.1) or a State Sequence Error (7.1; 11.6.1) is detected.

- (e) At the time a system including an MTM-Bus implementation is powered up, no S-module shall be addressed.
- (f) An S-module shall be considered "uniquely addressed" only after
- (i) a HEADER packet has been received that contains the S-module's module address, and the S-module is able to determine its module address unambiguously;
 - (ii) a HEADER packet has been received that contains the amnesia address, and the S-module is not able to determine its module address unambiguously (and is therefore responding to the amnesia address (3.3.1d(ii); 3.3.1h).

NOTE—Despite the fact that there may be multiple S-modules that can be addressed by the amnesia address, they all will behave as if they are uniquely addressed. In particular, they will return data if so required by a command addressed to 'FA' HEX.

- (g) An S-module shall respond to commands when and only when it is addressed.

NOTES

1—This rule (taken together with 3.3.1d) implies that an S-module that *can determine its module address unambiguously* can never execute commands sent to the amnesia address ('FA' HEX).

2—An S-module may take its MTM-Bus interface off-line during Start-up Built-In Test (SBIT) (18.2.1g) for local testing of the interface circuitry (the only time this is permitted). This means that the S-module might not receive commands sent during the self-testing period. The M-module may poll an S-module, requesting an ACKNOWLEDGE packet, to determine when an S-module is back on-line after self-testing and ready to receive MTM-Bus commands.

Recommendations

- (h) An S-module should be able to detect the condition of being unable to determine its module address unambiguously.

NOTE—To utilize the safety features supplied by the amnesia address, this recommendation has to be implemented. During the development of this Standard, addressing errors were observed in systems using prototype versions of this Standard; these errors could have been avoided and related faults detected had the capability of 3.3.1h and the amnesia address been specified at the time and implemented. Alternative methods of addressing security (such as appropriate Hamming distance between addresses) may be tolerable in a sparsely populated MTM-Bus address space, but might have a disadvantage because of not being supported by a standard.

- (i) The format/encoding of the module address as set in a given S-module during system assembly should provide for single-bit error detection when the address is read.

NOTE—For example, to increase the probability of an S-module's detecting an error in the accessing of its module address, the module manufacturer could add an additional, parity bit to the module address.

- (j) The module address of an S-module with a programmable module address should be 'FA' HEX when it is purchased (i.e., prior to programming of the module address).

NOTE—Details of the manner in which an S-module's module address is programmed have to be fully documented by the module's manufacturer (22.1.1b).

Permissions

- (k) Module addresses may be implemented by a variety of means, including

- (i) on-module firmware
- (ii) jumper wires
- (iii) DIP switches
- (iv) discrete pins on the backplane

3.3.2 Description

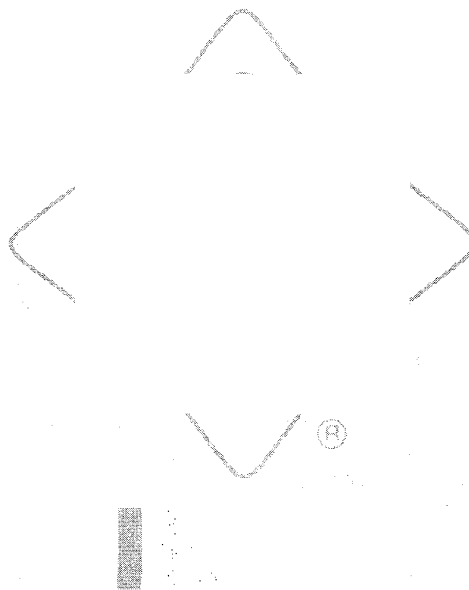
Each S-module is to be assigned an eight-bit address. The use of eight bits allows for a set of 256 addresses including 250 single module addresses ('0' through 'F9' HEX), one amnesia address ('FA' HEX) for use by an S-module that has lost access to its module address, one broadcast address ('FB' HEX), and four multicast addresses ('FC', 'FD', 'FE', and 'FF' HEX). An S-module is considered to be addressed (and thus allowed to execute the current MTM-Bus command) if it has received (in the most recently transmitted HEADER packet) its module address (or the amnesia address when appropriate), the broadcast address, or the multicast address that matches its current multicast select group. An S-module is considered uniquely addressed if it has received its module address or the amnesia address in the most recently transmitted HEADER packet. Whether an S-module is "addressed" or "uniquely addressed" influences the manner in which it responds during an MTM-Bus message.

If an S-module is unable to access its module address (e.g., due to a module address parity error, etc.), the S-module in question can respond only to commands sent to the amnesia address ('FA' HEX) or to any command sent to the broadcast address or the address of the S-module's current multicast group. The amnesia address provides the possibility of limited system diagnosis based on data recovered from S-modules unable to access their unique module addresses. Some data can be recovered from such modules using the amnesia address because the modules will act as though uniquely addressed. Diagnosis might begin by directing the M-module to read the module status of the module with address 'FA' HEX. If there is an "all zero" response, then no S-module is responding and, hence, unless more serious failures have occurred, no S-module has self-diagnosed an address fault. If multiple S-modules respond to 'FA' HEX (i.e., become addressed by a given message) and an ACKNOWLEDGE packet is required, the collision-error handling of the S-modules (11.7.1) can (at least sometimes) reduce the number of modules that remain addressed after ACKNOWLEDGE packet transmission to as few as one.

Note that at any given time all modules *not* in the group that respond to the amnesia address may be segregated in one multicast group. Then this group's address may be used as a substitute for the broadcast address. To simplify software design, it may be wise to have such a multicast group address established as

equivalent to the broadcast address from the outset and simply remove amnesiac modules to another multi-cast group if and when they are detected.

Each S-module has to respond to at least one of the 250 module addresses (or to the amnesia address) so that it can be addressed by the M-module. An S-module's unique module address may be provided to the MTM-Bus test interface on the S-module by a variety of means including firmware, jumper wires, dual in-line package (DIP) switches, or discrete wires to the backplane. With any of these methods, an additional address parity bit should be provided to ensure that a given S-module's MTM-Bus test interface will be able to detect the occurrence of single bit address faults and, consequently, will not respond to another S-module's address in the event of such a fault.



4. MTM-Bus Physical Layer requirements

The MTM-Bus Physical Layer defined in this clause includes the definition of the electrical and timing requirements for the MTM-Bus signals. Precise electrical and timing requirements are not specified to allow for selection of bus transceivers most appropriate to a specific application.

In this and subsequent descriptions of MTM-Bus layer requirements, the requirements are divided into three categories: Those having to do with S-modules, those having to do with M-modules, and those that apply to both module types. The intent of this division is to make it easier for a potential designer/manufacturer of modules to separate the requirements for M-modules and S-modules.

4.1 Physical layer requirements independent of module type

4.1.1 Specifications

Rules

- (a) All MTM-Bus signal drivers, excluding the driver of the MCLK signal, shall support a logical-OR operation such that if any module drives a logic 1 on such a signal, that signal will be forced to a logic 1.

NOTE—The MTM-Bus signals MCTL, MMD, MSD, and MPR may collectively use active-low/negative logic levels on the backplane. Such an implementation may be necessary in some technologies to obtain a logical-OR operation.

- (b) All MTM-Bus signal receivers shall interpret a logic 1 as actively driven.
- (c) Signal drivers for all MTM-Bus signals shall be designed so that the simultaneous driving of logic 0 and logic 1 values by different modules on a given MTM-Bus signal shall not cause physical damage to any driver of that signal.

Recommendations

- (d) The MTM-Bus signal drivers and receivers on a given MTM-Bus module should exhibit a high input impedance when that module is insufficiently powered for normal operation.

NOTE—This specification is a recommendation rather than a rule because there may be cases in which it is expected that all modules in a given implementation are always powered up. However, if this is not the case or if protection is desired in case of power supply failure, then following this recommendation may be very important.

4.1.2 Description

Logical-OR operation: Logical-OR operation on bus signals is required for a variety of reasons. The MSD signal requires logical-OR operation to make use of Contend for Bus command (16.4.1). Logical-OR operation of the MPR signal is needed to allow one or more of multiple S-modules to request additional time in the PAUSE Slave Controller state during a broadcast or multicast message so that the slowest S-module may request an adequate number of cycles in the PAUSE Slave Controller state (8.3). Logical-OR operation on the MMD and MCTL signals may be used to allow transfer of MTM-Bus mastership from one M-module to another without potentially damaging collisions.

Selecting bus transceivers: The MTM-Bus Physical Layer may be customized to use bus transceivers from any number of logic families such as open-collector transistor-transistor logic (TTL), backplane transceiver logic (BTL), and emitter coupled logic (ECL). The selection of a specific logic family may be influenced by

the length of the backplane, the number of expected loads, the bus frequency, noise immunity, cost, power, and available power supplies.

Logic levels: This MTM-Bus Protocol Standard does not provide detailed electrical and timing parameters for the module. Timing diagrams depict logic levels rather than voltage levels.

The bus signals MCTL, MMD, MSD, and MPR at the module interface may be implemented in a negative logic technology that supports logical-OR operation. Any modules that are intended to interoperate have to address the appropriate electrical parameters, including the logic level for these bus signals.

It is possible that the MCLK will be implemented in a different device technology than the other bus signals. The reasons are twofold:

- MCLK does not require logical-OR operation, and
- the other bus signals may experience ringing prior to settling before a clock transition.

The MCLK signal has to make a clean transition through its threshold region to avoid inadvertent data driving or latching. The MCLK logic level selection for the device technology may be based on an acceptable noise margin to avoid driving data inadvertently. A family of interoperable modules have to address the appropriate MCLK electrical parameters, including the logic levels.

4.2 M-module Physical Layer requirements

4.2.1 Specifications

Rules

- (a) The MMD and MCTL signal drivers of an M-module shall be active continuously during operation of the MTM-Bus.
- (b) An M-module shall provide collision/error detection circuitry on both MMD and MCTL signals. For each of these signals, the relevant circuitry in a given module, *M*, shall detect cases in which *M*'s MTM-Bus driver of the signal is attempting to force the signal to the logic zero state when the logical-OR function of the bus (4.1.1a) is resulting in a logic one value for the signal.
- (c) The M-module shall have the capability to detect the value of, and respond to, the MPR signal.
- (d) The M-module shall have the capability to detect the value of, and respond to, the MSD signal.

4.2.2 Description

See discussion of logical-OR operation in 4.1.2.

See discussion of collision-detection circuitry design options in 4.3.2.

4.3 S-Module Physical Layer requirements

4.3.1 Specifications

Rules

- (a) S-modules shall provide collision/error detection circuitry on the MSD signal connection. The relevant circuitry in a given module, *M*, shall detect cases in which *M*'s MTM-Bus driver of the MSD signal attempts to force the signal to a logic zero when the logical-OR function of the bus (4.1.1a) is resulting in a logic one value for the signal.

NOTE—Collision-detection circuitry plays a key role in the Contend for Bus (16.4.1) and Verify BMR (16.12.1) sequences.

- (b) The collision/error detection function on the MSD signal of a currently addressed S-module shall be enabled to set the SDF bit of the Bus Error register and release the MSD signal (11.7.1a) only if the S-module is uniquely addressed and actively transferring a packet required in response to the current command with the exception of the ACKNOWLEDGE packet (5.3) being transferred in response to a Contend for Bus command (16.4.1) or a Verify BMR command (16.12.1).

NOTE—Timing constraints on collision-detection circuitry are implied by 11.7.1a. Namely, it has to be possible to detect a collision and release the MSD signal before the S-module in question can transmit another bit on MSD—before the first rising edge of the MCLK signal following the rising edge of the MCLK signal on which the collision occurred.

Recommendations

- (c) The MPR signal should be implemented on all S-modules.

4.3.2 Description

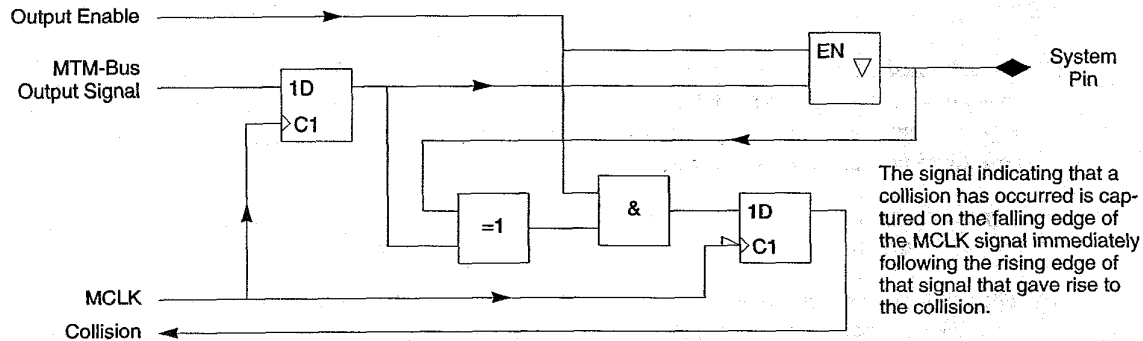
Collision detection: Collision detection is needed both to detect bus errors and to allow an S-module to release its MSD signal when it loses Contend for Bus operations (16.4.1) or Verify BMR operations (16.12.1). As illustrated in figure 4-1, collision detection may be implemented by providing a receiver for each output driver and comparing the (internal module) value to be output on the bus and the current value on the bus. This function has to be gated such that collisions are only reported when the output of the driver is enabled and the packet being transferred is not the ACKNOWLEDGE packet transferred in response to a Contend for Bus command.

It is possible to comply with the IEEE Std 1149.5 requirements for collision detection while using discrete bus transceivers or electrical level translators. As illustrated in figure 4-2, the component that provides support for the collision detection may have two discrete pins for each bus output. Separate electrical-level translators can then be connected to provide the logical-OR drive and to sense the actual bus value. With this approach, a single component that supports the MTM-Bus protocol, perhaps with TTL electrical levels, could be used with off-the-shelf transceivers to support open-collector TTL, ECL, BTL, or other backplane electrical levels.

See discussion of logical-OR operation in 4.1.2.

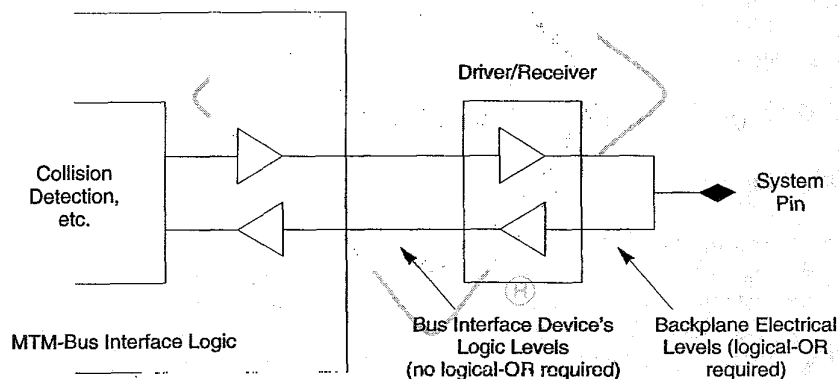
When an uniquely addressed S-module is transmitting a packet in response to the command contained in the current message [other than an ACKNOWLEDGE packet being sent in response to a Contend for Bus command (16.4.1)], the MTM-Bus interface circuitry of the S-module has to record detection of a collision on the MSD signal by setting the SDF bit of the S-module's Bus Error register (9.3).

Taking drivers off-line: When an S-module is involved in start-up self-testing it may take its MTM-Bus interface off-line (18.2.1g).



NOTE—This example circuit compares the logic value being driven with the logic value currently on the bus. A box containing “=1” is the IEEE standard symbol for an XOR gate.

Figure 4-1—Collision detection



NOTE—Separate off-the-shelf drivers and receivers can be used to customize a single MTM-Bus interface device for multiple Physical Layer protocols.

Figure 4-2—Bus transceivers

4.4 MTM-Bus timing requirements

The following symbols are used in specifying timing parameters and as labels in figures 4-3 through 4-5:

- t_0 is the time that MCLK is logic 0 per cycle
- t_1 is the time that MCLK is logic 1 per cycle
- T_1^M is the minimum operable t_1 for an MTM-Bus module or MTM-Bus interface device, M
- T_0^M is the minimum operable t_0 for an MTM-Bus module or MTM-Bus interface device, M

t_{LH} is the duration of MCLK rising edge transition
 t_{HL} is the duration of MCLK falling edge transition
 t_c is the MCLK cycle time ($t_{LH} + t_1 + t_{HL} + t_0$)

NOTE—The transition times t_{LH} and t_{HL} can be determined only for a given system. These times are not considered a part of t_1 or t_0 and have to be accounted for in the selection of t_c .

T_c^M is the minimum operable t_c for an MTM-Bus module or MTM-Bus interface device, M
 T_{LH}^M is the minimum operable t_{LH} for an MTM-Bus module or MTM-Bus interface device, M
 T_{HL}^M is the minimum operable t_{HL} for an MTM-Bus module or MTM-Bus interface device, M
 t_p is the maximum system backplane propagation and settling time
 $t_{cs}^{M,N}$ is the clock skew as measured between the MTM-Bus signal driver, M , and the MTM-Bus signal receiver, N
 T_{co}^M is the clock-to-output time for an MTM-Bus signal driver, M
 T_s^M is the set up time for an MTM-Bus signal receiver, M
 T_h^M is the hold time for an MTM-Bus signal receiver, M

Note that timing parameters of MTM-Bus modules, MTM-Bus interface devices, and individual drivers on MTM-Bus modules are designated in a T_z^M format. Note that there is a value of T_z^M specific to each of MPR, MMD, etc., for a given module. Timing parameters for a given system under discussion are in the t_y format.

4.4.1 Specifications

Rules

- (a) The MTM-Bus Clock (MCLK) signal shall be a single-phase clock.
- (b) In the implementation of the MCLK signal, a logic one shall be the more positive of the two voltages; and a logic zero, the least positive.
- (c) The manufacturer of an MTM-Bus module or MTM-Bus interface logic, M , shall document T_c^M (minimum operable value for t_c), T_1^M (minimum operable value for t_1), and T_0^M (minimum operable value for t_0).
- (d) An MTM-Bus module or MTM-Bus interface device, M , shall be compliant to this Standard for all values of t_c , t_1 , and t_0 where $t_c \geq T_c^M$, $t_1 \geq T_1^M$, and $t_0 \geq T_0^M$.
- (e) Stored-state devices contained in MTM-Bus interface logic shall retain their states indefinitely when the signal applied to MCLK is stopped at a logic 0 or a logic 1.
- (f) MTM-Bus signal outputs shall change state only following a *rising edge* of the MCLK signal.
- (g) All MTM-Bus signal inputs shall be latched following each *falling edge* of the MCLK signal.
- (h) The manufacturer of an MTM-Bus module or MTM-Bus interface device including an MTM-Bus signal driver, M , shall document T_h^M , T_{co}^M (figure 4-4), and T_s^M (figures 4-3 and 4-4).

Recommendations

- (i) For any MTM-Bus signal driver, M , the module timing parameters T_{co}^M , T_s^M , and T_h^M should be minimized.

NOTE—This will support performance maximization for an MTM-Bus implementation to which the given module is connected.

- (j) In any MTM-Bus module, M , the module timing parameter T_0^M should be minimized.

NOTE—This will allow the MCLK signal for systems containing the given module to be held = logic 1 for as long as possible to support secure backplane propagation.

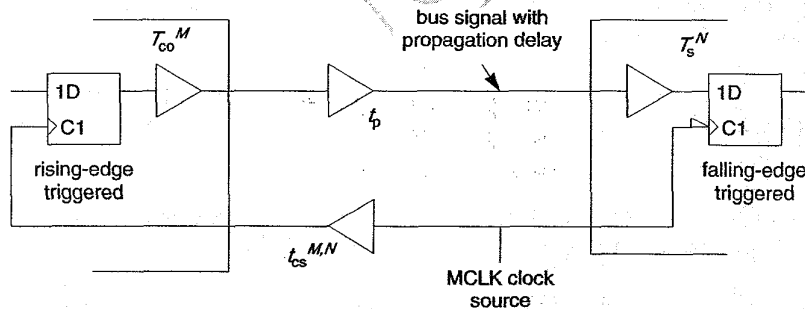
Permissions

- (k) The manufacturer of an MTM-Bus module, may impose maximum operable values for t_{LH} and t_{HL} .

4.4.2 Description

To ensure race-free operation, a dual edge clocking approach is used in this Standard. As illustrated in figure 4-3, an MTM-Bus signal driver (M) transmits data onto the bus a clock-to-output time (T_{co}^M) after the rising edge of MCLK. The data propagates and settles after the propagation and settling time (t_p). The data have to be valid at the receiving MTM-Bus signal input (N) at least setup time (T_s^N) prior to the falling edge of MCLK, that may *lead*, in the worst case, MCLK at the driver by $t_{cs}^{M,N}$. The module sourcing the data have to retain valid data at its relevant signal output until the end of the interval in which the MCLK signal is at logic 0. For all MTM-Bus signal driver/receiver pairs (M, N) such that M may drive a signal received by N , the relevant system timing relationship is characterized by

$$t_1 \geq T_{co}^M + t_p + T_s^N + t_{cs}^{M,N}$$



NOTE—An MTM-Bus signal output drives new data on the rising edge of MCLK, and an MTM-Bus signal input latches data on the falling edge of MCLK.

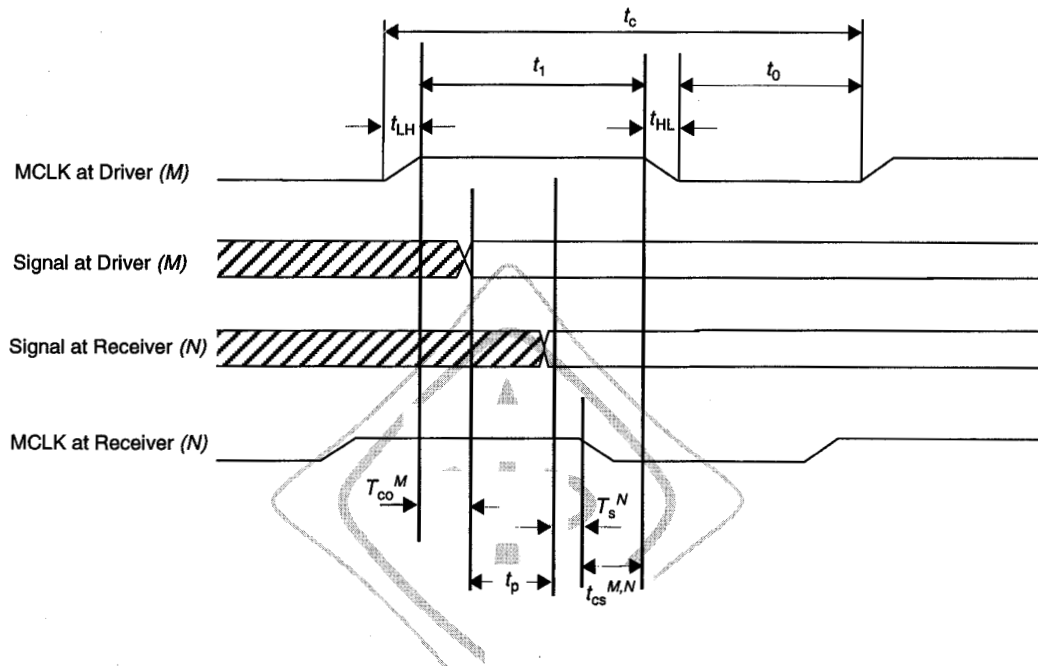
Figure 4-3—Data transfer

A timing diagram showing the MCLK and another MTM-Bus signal waveform at both a transmitter and a receiver are illustrated in figure 4-4.

To assess module hold requirements, consider the worst-case condition—that in which the clock at the receiver (MTM-Bus module N) *trails* the clock at the driver (MTM-Bus module M) as illustrated in

figure 4-5. The system parameter t_0 has to be chosen such that the falling edge does not latch new data at a time when the relevant signal value is changing. For any MTM-Bus signal output, M , and signal input, N , connected in a particular implementation of the MTM-Bus, the relevant system timing relationship is characterized by

$$t_0 \geq T_h^N + t_{cs}^{M,N}$$



NOTE—This timing diagram shows the transitions of MCLK and another MTM-Bus signal at both the signal driver (on MTM-Bus module M) and signal receiver (on MTM-Bus module N) in the worst-case propagation situation—where the clock at the receiver leads the clock at the source. In every bidirectional data transfer, operation in one of the directions will fit the worst case just described.

Figure 4-4—Data transfer timing

For a given MTM-Bus implementation in a given system, the MCLK waveform will have to be adjusted so that worst-case timing requirements for the connected MTM-Bus interface devices [equation (1)], the worst-case system propagation budget [equation (2)], and the worst-case module hold time requirement [equation (3)] are not violated.

$$t_c \geq \max (T_c^X : X \text{ is a module or interface device connected to the bus}) \quad (1)$$

$$t_1 \geq \max (T_{co}^X + t_p + T_s^Y + t_{cs}^{X,Y} : X \text{ is a signal output and } Y \text{ is a connected signal input}) \quad (2)$$

$$t_0 \geq \max (T_h^Y + t_{cs}^{X,Y} : X \text{ is a signal output and } Y \text{ is a connected signal input}) \quad (3)$$

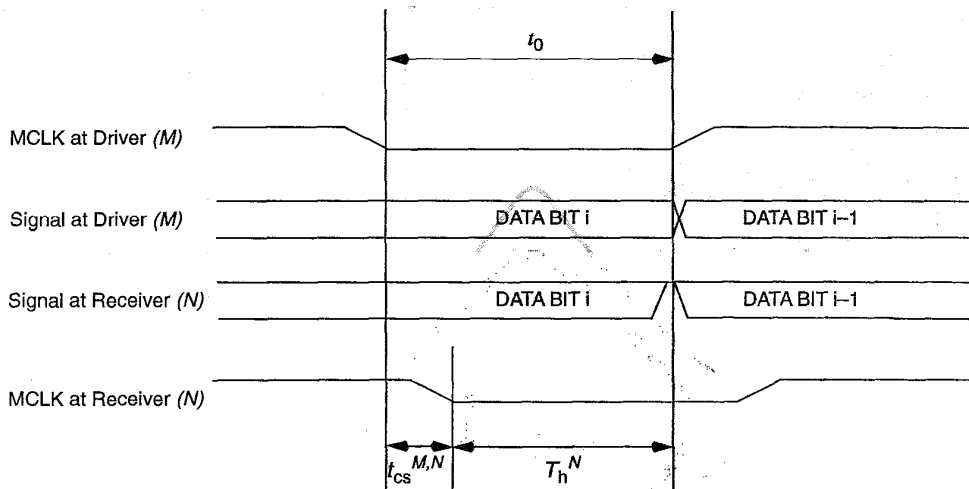
In addition, the system timing parameters cannot violate the worst-case module timing parameters.

$$t_1 \geq \max (T_1^M : M \text{ is a module connected to the bus})$$

$$t_0 \geq \max (T_0^M : M \text{ is a module connected to the bus})$$

$$t_c \geq \max (T_c^M : M \text{ is a module connected to the bus})$$

As long as the system MCLK waveform parameters t_1 , t_0 , and t_c meet the relationships described in the present clause, system timing requirements will be met. This method for specifying the MCLK waveform requires the bus interface to operate over a range of MCLK frequencies from as low as may be desired by a user to a documented maximum.



NOTE—This MTM-Bus timing diagram shows the transitions of the clock and another MTM-Bus signal at both the signal driver (M) and signal receiver (N) in the worst-case hold time situation. In this case, the clock at the receiver trails the clock at the source, propagation delay is zero, and clock to output delay for M is zero. The time t_0 has to equal or exceed the sum of the clock skew and the hold time of N .

Figure 4-5—Hold time requirement

Performance requirements: Although minimum performance requirements for an MTM-Bus module, M , are not specified, it is recommended that T_1^M and T_0^M be minimized to allow greater bandwidth if required by a system. An MCLK waveform has to be observed on the system backplane to ensure that module T_1^M and T_0^M requirements are met.

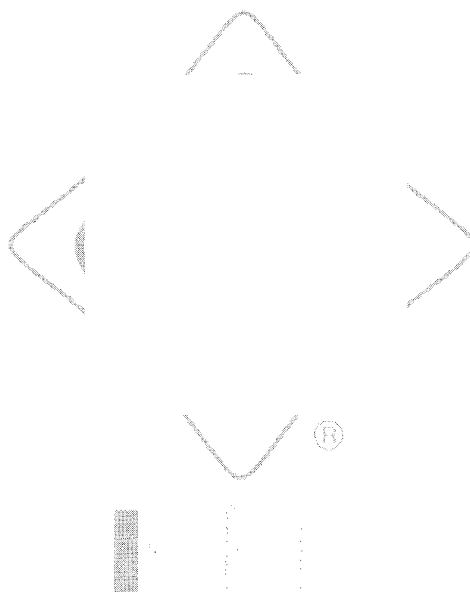
Integration risk: All MTM-Bus timing relationships involve t_1 , t_0 , and t_c . Thus decreasing the MCLK frequency (thereby increasing t_1 , t_0 , and t_c) will alleviate potential timing violations. In addition, the MTM-Bus dual-edge timing approach encourages design and use of MTM-Bus modules employing technologies that support “low” values of T_{co}^M , because this will tend to increase the backplane propagation budget.

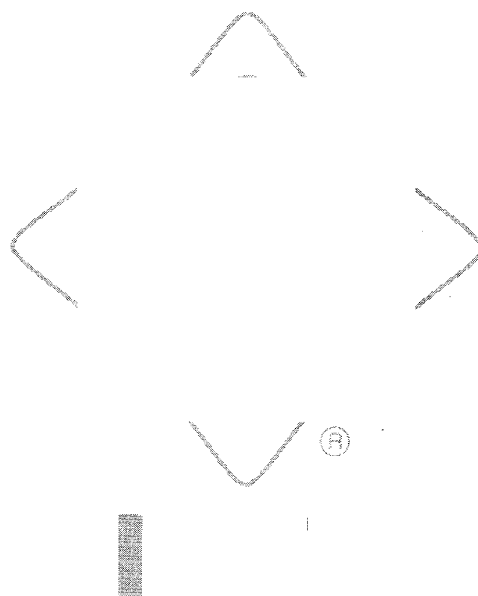
NOTE—The dual-edge timing approach was adopted over a single-edge approach. In a single-edge approach, data are driven and latched on the same signal edge. In the worst case, the clock skew between a transmitting module, M , and receiving module, N — $t_{cs}^{M,N}$ —has to be controlled to avoid latching data from a signal that is in transition. This condition is characterized by $t_{cs}^{M,N} \leq T_{co}^M - T_h^N$. Because the clock skew requirement in a single-edge system is independent of MCLK frequency, skew problems cannot be alleviated by reducing MCLK signal frequency. Implementations that reduce T_{co}^M place tighter requirements on clock skew. In a dual-edge approach, worst-case module and backplane timing constraints may always be alleviated by increasing t_1 or t_0 (at the cost of reducing MTM-Bus bandwidth).

Interoperability: Because this Standard addresses module timing parameters while leaving the system parameters flexible, a set of modules with compatible electrical characteristics may be interoperable in a number of systems having differing backplane timing parameters. The MCLK waveform can be selected to meet the backplane propagation budget and the capabilities of the selected MTM-Bus modules.

NOTE—The SAE Avionics TM-Bus Interoperability Standard AS4765 has adopted this timing approach and is adding additional detailed timing and electrical requirements to support development and deployment of interoperable modules in the avionics environment.

Support for debugging an integrated system: Stored state devices contained in the MTM-Bus interface logic have to retain their state even when the MCLK signal is stopped (in either a “0” or “1” state) as detailed by rule 4.4.1e. This requirement is made to allow debugging in a system integration environment in which the MCLK signal may be stopped for some reason (e.g., to permit an ATE test pattern load or to check signal values during error tracing). If state information or status/error information in stored state devices in the MTM-Bus interface logic were not retained when the signal on MCLK is stopped, the debugging process would be severely complicated.





5. MTM-Bus Link Layer: Packet requirements

All MTM-Bus messages consist of the transfers of 17-bit packets. The packet types include HEADER packet, ACKNOWLEDGE packet, PACKET COUNT packet, and DATA packet. Relation between, and ordering of, packets within a message as well as intervals between packets are treated in 6.1.1k, 6.1.1o, 6.1.1p, 8.3.1f, and clauses 12 through 20.

5.1 Requirements applicable to all packets

5.1.1 Specifications

Rules

- (a) In the context of this Standard, a packet shall be a sequence of 17 bits transferred between the M-module and one or more S-modules or from an S-module to the M-module.
- (b) Bit <0> of each packet shall contain the packet parity bit.
- (c) The parity of each packet shall be odd.

NOTE—The modulo-2 sum of the bits in a packet (bits <16..0>) has to be 1.

- (d) Each packet shall be serially transferred, msb first.
- (e) Each packet shall be of one of the following types: HEADER packet, ACKNOWLEDGE packet, PACKET COUNT packet, DATA packet, or NULL packet.

5.2 HEADER packet requirements

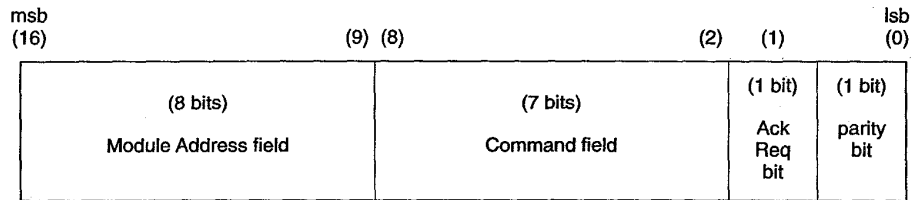
5.2.1 Specifications

Rules

- (a) The HEADER packet shall consist of the Module Address, Command, Acknowledge request, and parity fields as shown in figure 5-1.
- (b) The Module Address field shall be eight bits in length (bits <16..9>) and shall contain an S-module address (3.3.1).
- (c) The Command field shall be seven bits in length (bits <8..2>) and contain the 7-bit code determining the command to be executed by the addressed S-module(s).

NOTE—MTM-Bus commands are described in clauses 15 through 20.

- (d) The Acknowledge Request field shall be bit <1> and shall be used to request an ACKNOWLEDGE packet to be transmitted by any currently addressed S-module and to indicate that a PACKET COUNT packet will be the next packet transmitted by the M-module (14.1.1b).

**Figure 5-1—HEADER packet format**

5.2.2 Description

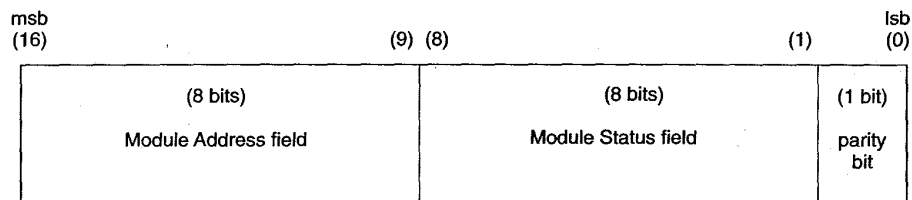
Precisely one HEADER packet occurs in each MTM-Bus message and is the first packet of the message. From the information in a HEADER packet, each S-module connected to an MTM-Bus can determine if it is addressed, what command is to be executed, and whether the system application controlling the M-module is requesting an ACKNOWLEDGE packet (5.3) from the S-module as part of the message transmission.

5.3 ACKNOWLEDGE packet requirements

5.3.1 Specifications

Rules

- The ACKNOWLEDGE packet shall consist of the Module Address, Module Status, and parity bit fields, as shown in figure 5-2.
- The Module Address field shall be 8 bits in length (bits <16..9>) and shall contain the address of the responding S-module.
- The Module Status field shall be 8 bits in length (bits <8..1>) and shall contain data residing in the Slave Status register prior to any changes to such data resulting from the receipt of the immediately preceding HEADER packet (9.2).

**Figure 5-2—ACKNOWLEDGE packet format**

5.3.2 Description

An ACKNOWLEDGE packet is transmitted by an S-module upon request and is both a convenient way to recover internal status information from an S-module and is designed to support the Contend for Bus sequence (16.4.1).

5.4 PACKET COUNT packet requirements

5.4.1 Specifications

Rules

- (a) The PACKET COUNT packet shall consist of a Packet Count field and odd parity bit as shown in figure 5-3.
- (b) The Packet Count field shall be 16 bits in length (bits <16..1>) and shall contain either the number of additional packet transfers remaining in the current message or an all-zeros pattern.
- (c) An all-zeros pattern in the Packet Count field shall indicate that the current message may contain any number of additional packets.

NOTE—Not knowing the length of a message in advance of receipt is not a problem. Termination of message transmission is controlled by the state of an fsm in the S-module, not by the contents of a packet of within a message (7.1; figure 7-1).

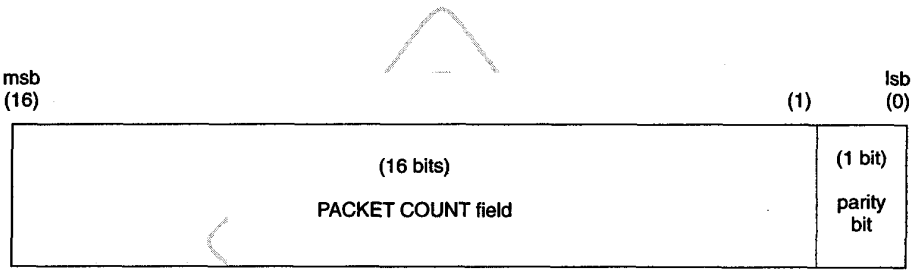


Figure 5-3—PACKET COUNT packet format

5.4.2 Description

If the Acknowledge Request bit is set in the HEADER packet of a message, the M-module will transmit a PACKET COUNT packet as the packet immediately following the HEADER packet. A PACKET COUNT packet conveys either the known number of packets still to be transmitted from M-module to S-module in the current message or the fact that the number of such packets has not been, or cannot be, predicted.

The information in a PACKET COUNT packet permits an S-module to determine if a data transfer has occurred as expected. An S-module will detect (9.3.1; figure 9-4; 11.11.1) when an ACKNOWLEDGE packet is requested and the message is terminated early (i.e., when no transfer of a packet takes place after the HEADER packet transfer or when a packet count is received, but does not match the actual number of packets received when message transmission is terminated). An S-module may also use the PACKET COUNT packet information to determine when a message is about to end or to flag an error if a message is too long or too short. Knowing when a message is about to end will be useful in implementing a message integrity check (e.g., CRC value, Hamming code, etc.) at the end of a message.

5.5 DATA packet requirements

5.5.1 Specifications

Rules

- (a) DATA packets shall consist of a Data field containing sixteen (16) bits (bits <16..1>) and one packet parity bit (bit <0>) as shown in figure 5-4.
- (b) Binary data 16 or fewer bits long that is to be transmitted in a DATA packet shall be placed in contiguous bit positions in the DATA packet such that the least significant bit (lsb) of the binary data is bit <1> of the DATA packet.

NOTE—Which bit of a bit string will be considered the lsb (msb) may be determined by the design(s) of the module(s) receiving a DATA packet or the design(s) of applications on that/those module(s) or the design of other system elements. This Standard mentions msb and lsb in this context to emphasize the necessity of the clear specification of expected bit order in a given module, application, or system. Manufacturers are required to document port-specific data formats (clause 21). Once the msb and lsb of a bit string are determined for a design, the placement of the resulting string in a DATA packet is then determined by the rules in 5.5.1.

- (c) If binary data to be transmitted in DATA packets consists of more than 16 bits,
 - (i) the data bit string shall be divided (beginning with its least significant 16 bits) into a sequence of 16-bit-long substrings (contiguous in the original data bit string) and, where necessary, a bit string (*S*) containing some number of bits (*n*, less than 16) representing the *n* most significant bits of the original data bit string;
 - (ii) the 16 least significant bits of the binary data shall be transmitted in the first transmitted DATA packet, and subsequent 16-bit-long substrings shall be transmitted (in order from least significant position to most significant position with regard to their order in the original data bit string) until all 16-bit-long substrings have been transmitted;
 - (iii) when all 16-bit-long substrings have been transmitted, then if there is a remaining substring to be transmitted [i.e., *S* of 5.5.1c(i)], then a final DATA packet shall be transmitted. This final DATA packet shall contain the substring *S* placed to fill the *n* least significant bits of the DATA packet's Data field.

NOTE—Given a substring that is constructed from an original data bit string to be transmitted over the MTM-Bus, which bit of that substring will be considered the lsb (msb) may be determined by the design(s) of the module(s) receiving the DATA packets or the design(s) of application(s) on that/those module(s) or the design of other system elements. This Standard mentions msb and lsb in this context to emphasize the necessity of the clear specification of expected bit order in a given module, application, or system. Manufacturers are required to document port-specific data formats (clause 21). Once the msb and lsb of a bit string are determined for a design, the placement of the resulting string in a DATA packet is then determined by the rules in 5.5.1.

- (d) The contents and format of the DATA packets transmitted as part of a given message shall be as defined for the command code contained in the Command field of the HEADER packet of that message.

NOTE—For example, see 17.1.1 (in particular, note documentation requirements) and the data transfers defined for the commands treated in clause 17.

Recommendations

- (e) Any bit positions within a DATA packet that do not contain a bit of a data bit-string being transmitted should be set to a logic 0.

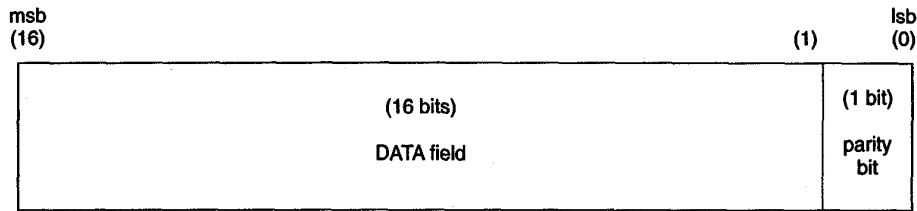


Figure 5-4—DATA packet format

5.6 NULL packet requirements

5.6.1 Specifications

Rules

- (a) A NULL packet shall be a DATA packet consisting of a 16-bit all-zeros field and an odd parity bit as shown in figure 5-5.

NOTE—To maintain odd parity, the parity bit of a NULL packet has to be set to logic one.

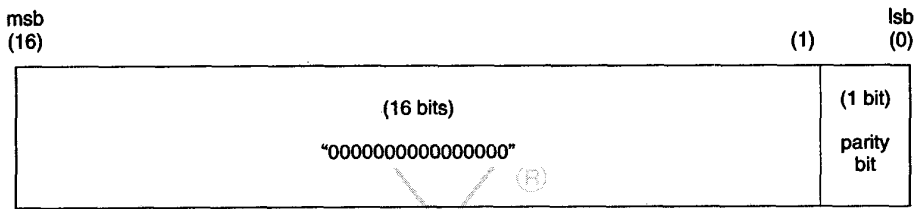


Figure 5-5—NULL packet format

5.6.2 Description

One use of a NULL packet is its transmission by the M-module during periods when one or more packets of data are being transmitted by an S-module while there is nothing to be transmitted in the reverse direction. The reverse case also will occur. It is important to note that, after transmission of a HEADER packet, during every period of a packet transfer during a singlecast, the MSD and MMD signals are being interpreted (by the M-module and addressed S-module, respectively) as conveying bits of a packet (4.2.1a; 6.1.1; 7.1.1). Therefore, failure to transmit a packet is never an option for an S-module addressed by a singlecast message; in such a situation, the transmission of a NULL packet with correct parity is necessary to operation of the MTM-Bus.

5.7 Formatting bit strings of more than 16 bits for transmission in DATA packets

Figure 5-6 provides an illustration demonstrating how a bit string of more than 16 bits is divided so that it can be appropriately transmitted in a series of DATA packets.

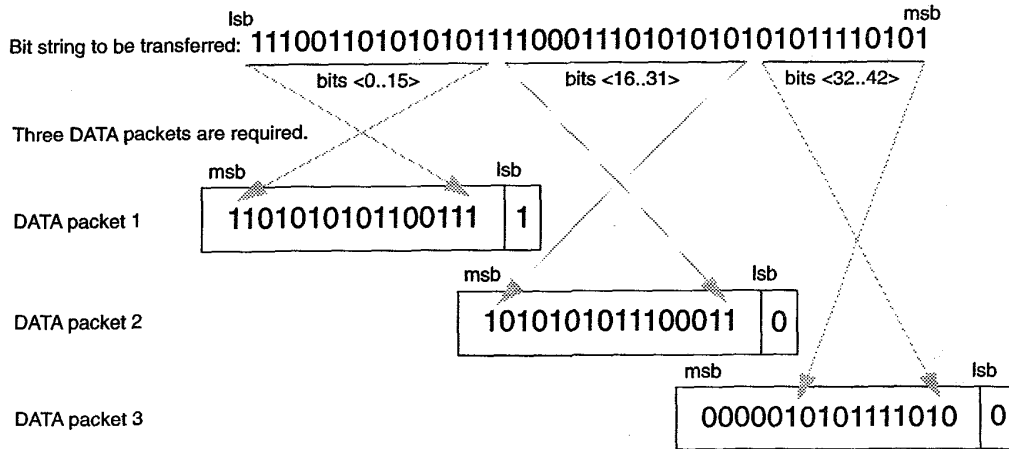


**Figure 5-6—Formatting a bit string of length greater than 16
for transmission in DATA packets**

In figure 5-6, the string to be transferred is presented with lsb left-most. Notice that DATA packets have the lsb in the right-most position in the figure. This means that each set of sixteen bits beginning with the lsb has to have its apparent order reversed when formatting the bit string for transmittal on the MTM-Bus in a DATA packet. The number of bits in the sample bit string is 11 modulo 16. Therefore, five logic zeros are added as padding in the third data packet; these padding bits are the five high order bits in that packet. In each MTM-Bus packet a parity bit (odd parity) is the lsb of the packet.