A Proposed Method of Accessing 1149.1 in a Backplane Environment

Lee Whetsel
Senior Member Technical Staff
Semiconductor Group

SCTA032

Reprinted with permission from Proceedings of International Test Conference,
Baltimore, Maryland, September 20–24, 1992.

A Proposed Method of Accessing 1149.1 in a Backplane Environment

Lee Whetsel
Texas Instruments Inc.
P.O. box 869305, M/S 8407
Plano, Texas 75086

Abstract

This paper presents a novel circuit and protocol that can be used in system backplanes to provide a connection method between backplane and board level IEEE 1149.1 serial buses. While the basic objective of this approach is to provide a simple 1149.1 backplane to board connection method, it can be expanded to include other features, some of which are similar to those being developed in the IEEE P1149.5 and P1394 backplane serial bus standards.

1.0 Introduction

The concept of a simple, cost-effective serial bus protocol providing linkage between 1149.1 board and backplane environments has been developed. This serial bus protocol operates on the backplane, using the signal wires defined in 1149.1, to address and select one of the boards in a backplane. Once selected, the board can be communicated to using the standard 1149.1 bus protocol. In general, the technique described in this paper can be applied to any type of bus. However, in this paper it is described as a feature added to the IEEE/ANSI 1149.1 standard serial bus designed for boundary scan testing of ICs at the board level [1]. The author assumes the reader has a basic understanding of the 1149.1 standard.

The approach described in this paper is based on a unique protocol. This protocol can be embedded in existing serial bus protocols and invoked, during times when the buses are in an idle or non-operational state, to address and select a slave device for access by a serial bus master. After the protocol has been used to connect a serial bus slave up to a serial bus master, it relinquishes control of the bus, and allows the bus to revert back to its normal mode of operation. One of the advantages of this approach is that it does not require modifying the protocol of the serial bus it is used with. Therefore, upgrading a preferred or existing serial bus to include the connection method provided by the protocol can be less costly and time consuming than a complete redesign approach.

The term "serial bus slave" used in this paper represents any logic module capable of being interfaced to and controlled by a serial bus master. While this papers describes serial bus slaves as being boards in a backplane, they could also be: sub-circuits in an IC, ICs on a multi-chip module, ICs on a board, backplanes in a subsystem, or subsystems in a system. The term "serial bus master" used in this paper represents any logic module capable of interfacing to and controlling a serial bus slave. While this paper will illustrate the serial bus master as existing on the backplane wiring, it actually exists either in a backplane board or as an external tester connected to the backplane.

2.0 Background

As the use of 1149.1 continues to grow, more interest is being focused on how to access 1149.1 board designs in a backplane environment. Several methods of accessing 1149.1 boards in a backplane environment have been proposed. The following is a brief description of each of these methods.

2.1 Using 1149.1 at the Backplane Level

The 1149.1 standard describes a 4-wire serial bus that can be used to transmit serial data between a serial bus master and slave device. The 1149.1 bus consists of a test mode select (TMS) signal, a test clock (TCK) signal, a test data output (TDO) signal, and a test data input (TDI) signal. The TMS and TCK signals are output from the master and input to the slave. The TDO output from the master is input to the TDI input of the slave, and the TDO output from the slave is input to the TDI input of the master. During serial access, the master outputs control on TMS and clock on TCK to allow serial data to be transferred between the master and slave, via the TDO and TDI bus connections.

While 1149.1 was developed to serially access ICs on a board, it can be used at the backplane level to serially access boards. 1149.1 has two serial access configurations, referred to as "ring" and "star", that can be used at the backplane level. The following describes both configurations and identifies problems with each when used at the backplane level.

2.1.1 1149.1 Backplane Ring Configuration

In a backplane 1149.1 ring configuration, all boards directly receive the TCK and TMS control outputs from a primary serial bus master (PSBM) and are daisy chained between the PSBM's TDO output and TDI input. During scan operation, the PSBM outputs control on TMS and TCK to scan data through all boards in the backplane, via its TDO and TDI bus connections. The problem associated with the ring configuration, is that the scan operation only works if all the boards are included in the backplane and are operable to scan data from their TDI input to TDO output. If one of the boards is removed or has a fault, the PSBM will be unable to scan data through the backplane. Since the ring configuration does not allow access to remaining boards when one is removed or disabled, it does not fully meet the needs of a backplane serial bus.

2.1.2 1149.1 Backplane Star Configuration

In a backplane 1149.1 star configuration, all boards directly receive the TCK and TDI signals from the PSBM and output a TDO signal to the PSBM. Also each board receives a unique TMS signal from the PSBM. In the star configuration only one board is enabled at a time to be serially accessed by the PSBM. When a board is enabled, the TMS signal associated with that board will be active while all other TMS signals are inactive. The problem with the star configuration is that each board requires its own TMS signal. In a backplane with 50 boards, the PSBM would have to have 50 individually controllable TMS signals, and the backplane would have to have traces for each of the 50 TMS signals. Due to these requirements, star configurations are typically not considered for backplane applications.

2.2 Interfacing 1149.1 to other IEEE Buses

Two IEEE serial bus standards, P1149.5 and P1394, are in development for use in system backplanes. Since these standards are being specifically designed for backplane applications, they overcome the problems stated using 1149.1 as a backplane bus. However, the protocols of these anticipated standards are different from the 1149.1 protocol and therefore methods must be defined to translate between them and 1149.1. The following sections describe each backplane bus and identify problems with each when used to interface into 1149.1 board environments.

2.2.1 Interfacing P1149.5 to 1149.1

The P1149.5 standard working group is defining a module test and maintenance bus that can be used in system backplane environments [2,3]. P1149.5 is a single master/multiple slave bus defined by a 5-wire interface. Two of the wires are used for transferring serial data between the bus master and slave devices, one wire is used as a clock, one wire is used to control the operation of the bus, and one wire is used as a pause request from a slave to the master. The P1149.5 bus master initiates a data transfer operation by transmitting a data packet to all slave devices. The data packet consists of an address and command section. The slave device with a matching address is enabled to respond to the command section of the data packet as described in the P1149.5 standard proposal.

While the P1149.5 is a good data transfer type backplane bus for high-end commercial and military systems, its capabilities may exceed the requirements of some middle and low-end commercial systems that don't require or support its command set. Interfacing P1149.5 into an 1149.1 environment can be done but the system hardware and software designers must have an understanding of both bus types. One of the problems, therefore, in using P1149.5 to only interface into an 1149.1 environment, is that it adds an unnecessary complication to an otherwise simple serial access approach. Another problem is that the bandwidth of the 1149.1 serial data transfer may be adversely affected by the P1149.5 to 1149.1 protocol conversion process.

2.2.2 Interfacing P1394 to 1149.1

The P1394 standard working group is defining a 2-wire high-speed serial bus that can be used in either a cable or system backplane environment [4]. The P1394 standard, unlike P1149.5, is not a single master/multiple slave type bus. In P1394, all devices (nodes) connected to the bus are considered to be of equal mastership. Control of the bus is achieved by one node winning an arbitration contest with the other nodes in the network. Once a node wins control of the bus, it can transfer data to or from any other node in the network. The fact that P1394 can operate on a 2-wire interface makes this bus attractive in newer 32-bit backplane standards where only two wires are reserved for serial communication [5,6,7,8]. However, there are problems in using P1394 as a backplane test bus to access 1149.1 board environments.

The first problem is that P1394 is significantly more complex in operation than 1149.1, thus devices designed to translate between P1394 and 1149.1 may be costly. The second problem is that P1394 is not a full time test bus, but rather it is a general purpose serial communication bus. Its primary purpose in a backplane environment is to act as a backup interface in the event the parallel interface between boards becomes disabled. While 1149.1 test access can be achieved via P1394, it will be available only during time slices when the bus is not handling functional operations. Thus on-line 1149.1 test access will be limited and must be coordinated with other transactions occurring on the P1394 bus.

2.3 Extending 1149.1 for Backplane Usage

Another method of achieving a backplane to board level interface is to extend the protocol defined in the 1149.1 standard. Such an approach has been described in a paper presented at the 1991 International Test Conference by D. Bhavsar [9]. While the approach described in the paper has the same basic goal in mind as the one presented in this paper, they are fundamentally different in the method used to achieve the goal.

The Bhavsar paper describes a method of extending the protocol of 1149.1 to where it can be used to access an interface circuit residing between the backplane and board level 1149.1 buses. The interface circuit responds to 1149.1 protocol transmitted over the backplane bus to load an address. If the address matches the address of the interface circuit, the interface circuit is connected to the backplane. After the interface circuit is connected to the backplane, additional 1149.1 protocol is input to the interface circuit to connect the backplane and board level 1149.1 buses. Following this connection procedure, the board level 1149.1 bus can be controlled by the backplane 1149.1 bus. While Bhavsar's approach is an interesting one, it has one problem that limits its effectiveness as a general purpose 1149.1 backplane to board interfacing method.

The problem is that the approach does not allow for selecting one board, then selecting another board without first resetting the backplane and board level 1149.1 buses, by transitioning them into their test logic reset (TLRST) state. Entering the TLRST state causes test conditions setup in the ICs of a previously selected board to be lost due to the test reset action of the 1149.1 bus on the test access ports (TAPs) of the ICs.

For example, if the interconnects between two boards are to be tested, it is necessary to select and setup one of the boards to output a test pattern, then select the other board to receive the test pattern. With the described approach, the only way to select the second board, after the first board has been selected and setup, is to place the backplane serial bus in its TLRST state. The action of placing the backplane 1149.1 bus in it TLRST state clears out the test pattern setup in the first selected board, so the second selected board cannot receive the intended test pattern.

In another example, it may be desirable to select and initiate self-tests in a selected group of backplane boards. However, since the approach requires resetting the 1149.1 bus each time a new board is selected, it is impossible to self-test more than one board at a time, because resetting the bus aborts any previously initiated self-test.

3.0 A New Backplane Access Approach

The backplane access approach described in this paper provides a method of using the 1149.1 bus at the backplane level without incurring the problems previously described. Using this approach, it is envisioned that one homogeneous serial bus may be used throughout a system design, rather than translating between multiple serial bus types. Employing a common serial bus in system designs can simplify software and hardware engineering efforts, since only an understanding of one bus type is required.

A circuit, called an addressable shadow port (ASP), and a protocol, called a shadow protocol, have been defined to provide a simple method of directly connecting 1149.1 backplane and board buses together. When the 1149.1 backplane bus is in either its run test/idle (RT/IDLE) or TLRST state, the ASP can be enabled, via the shadow protocol, to connect a target board's 1149.1 bus up to the backplane 1149.1 bus. After the shadow protocol has been used to connect the target board and backplane buses together, it is disabled and becomes transparent to the operation of the 1149.1 bus protocol.

A board example using the ASP is shown in Figure 1. The board consists of multiple ICs and an ASP. The ICs operate, when connected to the 1149.1 backplane bus, via the ASP, as described in the 1149.1 standard. The ASP has a primary port for connection to the backplane 1149.1 bus, a secondary port for connection to the board 1149.1 bus, and an address input. The primary port signals are labeled; PTDI, PTDO, PTCK, and PTMS. The secondary port signals are labeled; STDI, STDO, STCK, and STMS. The address input to the ASP is a binary value used to identify the board on which the ASP mounted.

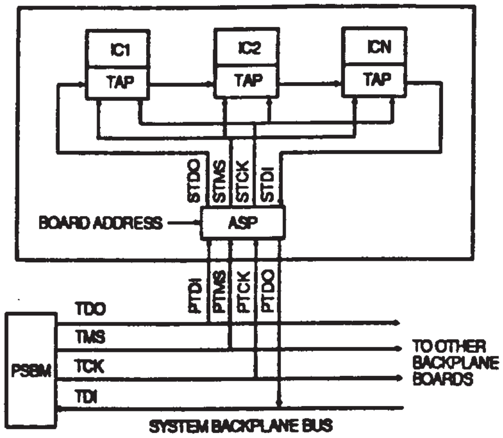

FIGURE 1 Board Using ASP Circuit

In Figure 2, multiple boards, similar the one in Figure 1, are shown interfaced to a PSBM via ASPs. When one of the boards needs to be accessed, the PSBM transmits a selection shadow protocol, called a select protocol, to address and enable the ASP of the selected board. The PSBM transmits the select protocol while the backplane 1149.1 bus is in either the RT/IDLE or TLRST state. The select protocol contains an address that is used to match against the address input to the ASP. All ASPs receive the select protocol, but only the one with the matching address is selected.

In response to the select protocol, the selected ASP transmits an acknowledgement shadow protocol, called an acknowledge protocol, to the PSBM to verify reception of the select protocol. The acknowledge protocol contains the address of the selected ASP to allow the PSBM to verify the correct ASP was selected. After transmitting the acknowledge protocol, the selected ASP makes a connection between its primary and secondary ports.

In response to the acknowledge protocol, the PSBM communicates to the selected board using the 1149.1 bus protocol. If the PSBM does not receive an acknowledge protocol, it assumes the board has been removed or is disabled and will not attempt to communicate to it using the 1149.1 protocol.

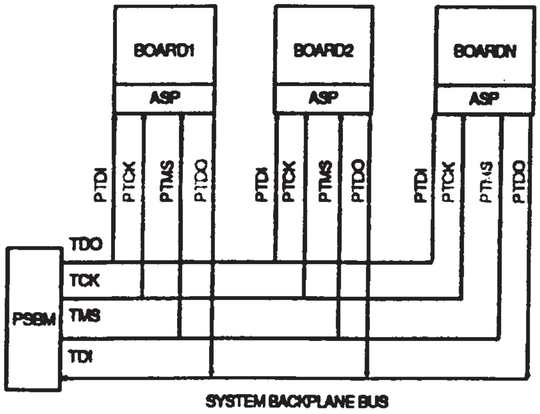

FIGURE 2 Backplane ASP Connections

After the PSBM completes its 1149.1 access of the currently selected board, it can output a new select protocol to select another board's ASP. In response to the new select protocol, the newly selected ASP transmits an acknowledge protocol back to the PSBM, then connects its primary and secondary ports. Also in response to the new select protocol, the previously selected ASP breaks the connection between its primary and secondary ports. The disconnecting ASP remains in the state the backplane 1149.1 bus was in when the disconnect occurs, i.e. the 1149.1 RT/IDLE or TLRST state. The ability to disconnect and leave a board level 1149.1 bus in the RT/IDLE state is very important since it allows leaving a board in a test mode while other boards are being selected and accessed.

A key objective in developing this backplane access approach was designing the select and acknowledge protocols so that they could be transmitted, via the 4-wire 1149.1 bus, without infringing upon the 1149.1 bus protocol. This objective was met by specifying that the select and acknowledge protocols could not use the 1149.1 TMS signal, and that the protocols could only be transmitted while the 1149.1 bus is idle in its RT/IDLE state or reset in its TLRST state.

In the RT/IDLE and TLRST states, TDO and TDI are disabled and pulled high (via pull-ups on TDI), TCK free runs, and TMS is held at either a logic zero or one state. While the 1149.1 bus is in one of these two states, the PSBM can output the select protocol from the PSBM's TDO output to the PTDI inputs of the ASPs, and receive the acknowledge protocol from the selected ASP's PTDO output on the PSBM's TDI input. Since the 1149.1 bus is inactive, the transmission of the select and acknowledge protocols is transparent to the 1149.1 bus, and does not infringe upon its protocol.

3.1 Design of Select/Acknowledge Protocols

To transmit the select and acknowledge protocols without using the TMS control signal, a bit-pair signaling method was designed to allow control and data to be transmitted together on a single wiring channel. During select protocols, the bit-pair signaling method allows the PSBM to transmit control and data from its TDO output to the ASP's PTDI input. During acknowledge protocols, the bit-pair signaling method allows the selected ASP to transmit control and data from its PTDO output to the PSBM's TDI input. Both protocols include control to indicate: an idle condition, a start data transfer condition, and a stop data transfer condition. In addition, both protocols include a method of transmitting data during the interval between the start and stop data transfer conditions.

The bit-pair signals are output from the transmitting device (PSBM or ASP) on the falling edge of the TCK and input to the receiving device (PSBM or ASP) on the rising edge of the TCK. Since this timing is consistent with 1149.1 timing, upgrading a PSBM to support this approach is simply a matter of forcing the TMS output to hold its present state ("0" for RT/IDLE and "1" for TLRST) while using normal 1149.1 scan operations to transmit and receive the select and acknowledge protocols. The simplicity of this approach makes it an attractive addition to the 1149.1 test bus. The bit-pair signals used in the select and acknowledge protocols are defined in the following list.

Idle Bit-Pair – an encoded control signal (I) identified by the transfer of two successive logic one bits from a transmitter to a receiver.

Select Bit-Pair – an encoded control signal (S) identified by the transfer of two successive logic zero bits from a transmitter to a receiver.

Logic 1 Bit-Pair – an encoded logic one signal (D) identified by the transfer of a logic zero bit followed by a logic one bit from a transmitter to a receiver.

Logic 0 Bit-Pair – an encoded logic zero signal (D) identified by the transfer of a logic one bit followed by a logic zero bit from a transmitter to a receiver.

3.2 Framing of Select/Acknowledge Protocols

A diagram of the select and acknowledge protocols being transmitted while the 1149.1 bus is in its RT/IDLE state is shown in Figure 3. The T signals shown in the protocol sequence indicate when the TDO to PTDI and PTDO to TDI wiring channels are tristate and pulled high. The first sequence framed between the first and second I signals is the select protocol output from the PSBM to the ASP (TDO to PTDI). The second sequence framed between the first and second I signals is the acknowledge protocol output from the selected ASP to the PSBM

(PTDO to TDI). The select protocol always precedes the acknowledge protocol as shown in the diagram.

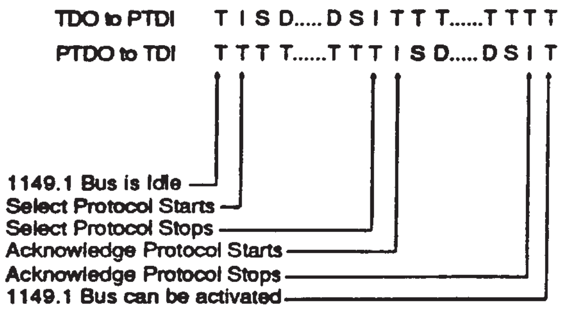
TDO to PTDI T I S D.....D S I T T T......T T T T
PTDO to TDI T T T T......T T T I S D.....D S I T
1149.1 Bus is Idle
Select Protocol Starts
Select Protocol Stops
Acknowledge Protocol Starts
Acknowledge Protocol Stops
1149.1 Bus can be activated

FIGURE 3 Select and Acknowledge Protocols

The I signal at the beginning of each protocol is designed to be indistinguishable from the preceding T signals. This avoids unintentional entry into a select or acknowledge protocol when the 1149.1 bus enters the RT/IDLE state after a scan operation. However, the I signal at the end of each protocol is designed to be distinguishable from the preceding S and D signals so that it can be used to terminate the protocol. Inside each protocol, first and second S signals are used to frame the address which is defined by a series of D signals. The logic zero and one D signals are distinguishable so that the binary address can be recovered.

3.3 ASP Circuit Description

A circuit example of the ASP is shown in Figure 4. The ASP consists of a receiver circuit (RCR), a transmitter circuit (XMT), a slave control circuit, multiplexers (MX1 and MX2), a power up reset circuit (PRST), and a reset address (RSTA). The primary port signals (PTDI, PTMS, PTCK, PTDO) connect to the backplane level 1149.1 bus. The secondary port signals (STDO, STMS, STCK, STDI) connect to the board level 1149.1 bus. The address input bus receives the board address.

3.3.1 ASP Receiver Circuit

The receiver circuit consists of a controller and a serial input/parallel output (SIPO) register. The PTDI signal from the PSBM is input to the receiver's SIPO register to supply the serial address during select protocols, and input to the receiver's controller to regulate the receiver during select protocols. The parallel address output from the SIPO is input to the slave control circuit via the address input (AI) bus. The status output from the receiver is input to the slave controller circuit to indicate when a select protocol has started, when the address is ready to read, and when the select protocol has completed. The control bus input to the receiver from the slave control circuit enables the receiver to respond to a select protocol input. The receiver is only enabled when the backplane 1149.1 bus is in the RT/IDLE or TLRST state.

The receiver's controller determines when a first "I-S-D" signal sequence occurs on PTDI, indicating the start of a select protocol and address input. In response to this input sequence, the controller enables the SIPO to receive the serial address input on PTDI. The controller determines when a first "D-S-I" signal sequence occurs on PTDI, indicating the end of the address input and select protocol. In response to this input sequence, the controller signals the slave control circuit, via the status bus, to read the address, then terminates the select protocol input operation.

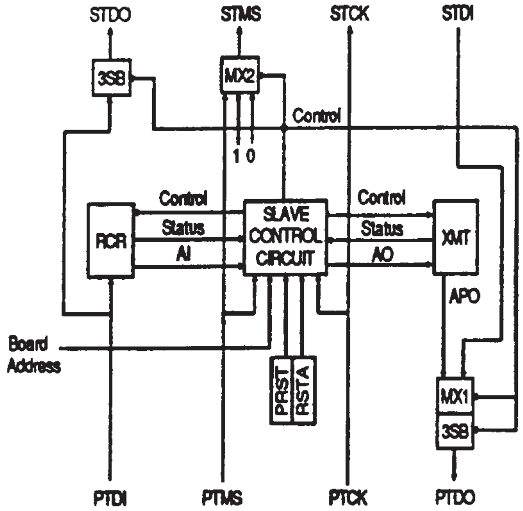

FIGURE 4 ASP Circuit Example

3.3.2 ASP Transmitter Circuit

The transmitter circuit consists of a controller and a parallel input/serial output (PISO) register. The transmitter's PISO register receives parallel data from the slave control circuit via the address output (AO) bus, and outputs the address serially to the PTDO output via the acknowledge protocol output (APO) signal and MX1. The transmitter's controller receives control input from the slave control circuit via the control bus, and outputs status to the slave control circuit via the status bus. The control input regulates the parallel to serial conversion process that takes place during the acknowledge protocol. The control input only enables the transmitter to output an acknowledge protocol when the backplane 1149.1 bus is in the RT/IDLE or TLRST state. The status output informs the slave control circuit of the transmitters status, i.e. whether the acknowledge protocol is in progress or complete.

At the beginning of an acknowledge protocol, the slave control circuit enables MX1 and the 3-state buffer (3SB) to pass the APO signal from the transmitter to the PTDO output. The slave control circuit then inputs the board address to the transmitter via the AO bus. In response to the address input, the transmitter outputs an I and S signal on PTDO to start the acknowledge protocol, then transmits the address on PTDO. After the address is shifted out, the transmitter circuit outputs an S and I signal sequence to stop the acknowledge protocol.

3.3.3 ASP Slave Control Circuit

The slave control circuit regulates the operation of the transmitter, receiver, and multiplexers during select and acknowledge protocols. The slave control circuit is clocked by the PTCK input from the primary port. The PTMS input from the primary port indicates to the slave control circuit when the 1149.1 bus is busy, idle or reset. While the backplane bus is idle (RT/IDLE) or reset (TLRST), the slave control circuit enables the transmitter and receiver circuits. The status buses from the receiver and transmitter circuits are used to input status to the slave control circuit. The AI bus from the receiver inputs the address received during select protocols. The AO bus from the slave control circuit outputs the board address to the transmitter during acknowledge protocols. The address input from the reset address (RSTA) allows resetting the ASP in response to a reset address input during a select protocol. The input from the power up reset circuit (PRST) allows resetting the ASP at power up.

During select protocols, the slave control circuit receives parallel address input from the receiver via the AI bus. The slave control circuit compares the received address against the board address. If the addresses match, the ASP responds by outputting an acknowledge protocol.

During the acknowledge protocol, the slave control circuit outputs control to the transmitter to load the board address and initiate the acknowledge protocol. After the acknowledge protocol has been transmitted, the slave control circuit outputs control to connect the primary and secondary ports.

3.4 Resetting the ASP

When power is first applied to the ASP, the slave control circuit is reset by input from the power-up reset circuit (PRST). When reset, the transmitter and receiver circuits are initialized and the primary and secondary ports are disconnected by disabling the STDO and PTDO outputs and setting the STMS output high. The STCK output always outputs the PTCK input. If desired, a reset input could be used to reset the ASP as well.

The ASP can also be reset by inputting a select protocol with an address that matches the reset address (RSTA) inside the ASP. If the address input matches the reset address, the ASP is reset to the same state as described in the power-up reset. The reset address is the same for all ASPs so that a global reset of all ASPs can be achieved by the transmission of a single select protocol containing the reset address. The reset address is unique from the board addresses. A preferred value for the reset address is zero, since board addressing will usually start with an address of one. An acknowledge protocol is not transmitted after a reset address has been received, to avoid contention on the PTDO outputs of multiple ASPs.

3.5 Disconnecting a Selected ASP

When 1149.1 access to another board is required, a new select protocol is issued from the PSBM. When the previously selected ASP receives the new select protocol its primary and secondary ports are disconnected. If the new select protocol was issued while the backplane 1149.1 bus was in its RT/IDLE state (PTMS=0), MX2 of the disconnecting ASP outputs a logic zero on STMS, to force the board level 1149.1 bus to remain in the RT/IDLE state. If the new select protocol was issued while the backplane 1149.1 bus was in its TLRST state (PTMS=1), MX2 of the disconnecting ASP outputs a logic one on STMS, to force the board level 1149.1 bus to remain in the TLRST state. Once again, the ability to maintain the RT/IDLE state on a disconnected board is very important because it allows tests to be setup and executed on more than one board at a time.

3.6 Advantages in Using ASPs

When comparing the described 1149.1 star configuration against the ASP configuration of Figure 2, it is clear that the ASP approach eliminates the need for additional TMS signals required by the star configuration. Thus the ASP provides a method of overcoming the problem stated for the 1149.1 star configuration.

Also, when comparing the use of different backplane buses to interface into 1149.1 board environments vs using the ASP, it is clear that the ASP does not require use of sophisticated, and bandwidth reducing translation circuitry. Thus the ASP provides a method of overcoming the problems related with using different backplane buses to access 1149.1 board environments.

Further, since the select and acknowledge protocols can be transmitted while the 1149.1 bus is in either the RT/IDLE or TLRST states, the board being disconnected can be left in either an idle or reset state. Thus the ASP provides a method of overcoming the forced-reset-on-disconnect problem associated with the approach described by Bhavsar.

4.0 Commandable ASPs

In small backplanes, a single centralized PSBM may be all that is necessary to serially access boards for test and maintenance operations. However, as the number and complexity of boards in a backplane grows, the serial access task increases to where a single centralized PSBM cannot handle the task in a timely manner. Anticipating the need for distributed test control, the ASP can be expanded to include a connection method and command set to enable board resident remote SBMs (RSBM) to autonomously test boards.

The addition of a command set, enables the ASP to perform other features in addition to its basic backplane to board connection function. Some of the commandable features include; (1) a method of connecting RSBMs to the board level 1149.1 bus, (2) a method of commanding RSBMs to independently test boards, (3) a method of non-intrusively monitoring the status of a remote test operation, and (4) a method of transferring data between a board resident memory and PSBM. The ASP's data transfer method achieves the same goal as the data transfer methods used in the P1149.5 and P1394 standard proposals. These commandable features further improve the ASPs ability to serve, in combination with 1149.1, as a system backplane test bus.

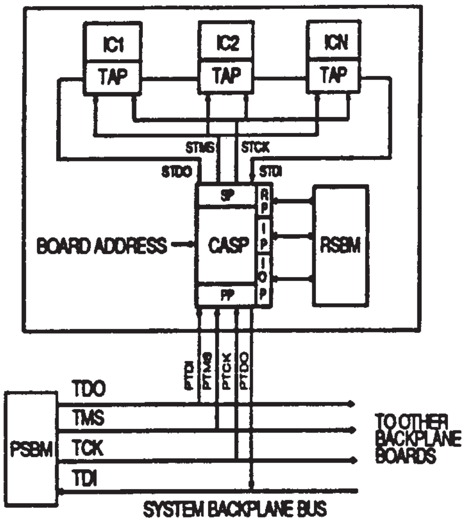

FIGURE 5 CASP Circuit Application

In Figure 5, a board is shown consisting of functional ICs (1-n), a commandable ASP (CASP), and a RSBM. The RSBM consists of a processor for executing local test programs, a memory for storage of test programs and data, an interrupt port, and an 1149.1 master interface port. The RSBM's processor and memory can either be dedicated for test or shared with the system logic on the board. The CASP consists of an 1149.1 primary port (PP) for interfacing to the backplane PSBM, an 1149.1 secondary port (SP) for interfacing to the functional ICs, an 1149.1 remote port (RP) for interfacing to the RSBM's master interface port, an interrupt port (IP) for interfacing to the RSBM's interrupt port, and an I/O port (IOP) for interfacing to the RSBM's memory. If required, additional I/O ports may be added to the CASP to provide parallel interfaces to other memories or I/O devices.

4.1 Expanded Select Protocol

To allow commands to be input to the CASP, the select protocol is expanded to allow for command transfer. In the ASP, a select protocol was defined by the transfer of a first I signal to start the select protocol, followed by the transfer of an address frame (of D signals) bounded by first and second S signals, followed by a second I signal to stop the select protocol. The select protocol of the CASP follows this format but expands the definition of the address frame into what is referred to as a message frame.

The select protocol message frame consists of a header containing an address (ADD) and command (CMD) field, and a cyclic redundancy check (CRC) field. The address field selects the CASP, the command field commands the CASP, and the CRC field is used for error detection. All fields within the message frame are separated by an S signal. The message frame may include optional fields between the header and the CRC field, as required by the command being sent. While the framing method allows fields to be transferred in either a fixed or variable D signal length, a fixed length field is preferred because it simplifies memory packing/unpacking operations, and improves error detection using simple signal counting techniques.

In Figure 6, examples of the two types of CASP select protocols are shown. Type 1 has a message frame containing the header's ADD and CMD fields, and the CRC field. Type 2 has a message frame containing the header's ADD and CMD fields, optional fields (OF) 1-N, and the CRC field.

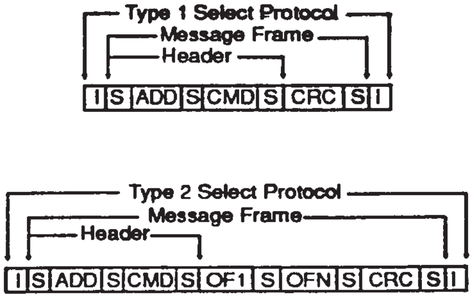

FIGURE 6 Expanded Select Protocols

In response to receiving either select protocol type (1 or 2) from the PSBM, the CASP checks its address against the received address field. If the addresses do not match, the CASP ignores the remainder of the select protocol and does not send an acknowledge protocol. If the addresses match, the CASP checks the command field against a set of known commands to see what operation is to be performed. In response to an unknown command, the CASP ignores the remainder of the select protocol, sets a command error bit in its status register, then sends an appropriate acknowledge protocol type (1 or 2) to the PSBM, to indicate the command error. In response to a known command, the CASP receives the remainder of the select protocol, then matches the received CRC field against a CRC it calculates on the data received in the select protocol. If the CRCs do not match, the CASP ignores the command, sets a CRC error bit in its status register, then sends an appropriate acknowledge protocol type to the PSBM, to indicate the CRC error. If the CRCs match, the CASP sends an appropriate acknowledge protocol type to the PSBM, to indicate that an error-free select protocol was received.

4.2 Expanded Acknowledge Protocol

To allow the PSBM to verify that the command input to the CASP was received correctly, the acknowledge protocol is expanded to allow for status transfer. In the ASP, the acknowledge protocol was defined by the transfer of a first I signal to start the acknowledge protocol, followed by the transfer of an address frame bounded by first and second S signals, followed by a second I signal to stop the acknowledge protocol. The acknowledge protocol of the CASP follows this format but expands the definition of the address frame into a message frame.

The acknowledge protocol message frame consists of a header containing an address (ADD) and status (STS) field, and a CRC field. The address field identifies the CASP, the status field informs the PSBM of the CASP status, and the CRC field is used for error detection. All fields within the message frame are separated by an S signal. The message frame within the acknowledge protocol may include optional data fields between the header and the CRC field, if required by the command sent in the previous select protocol.

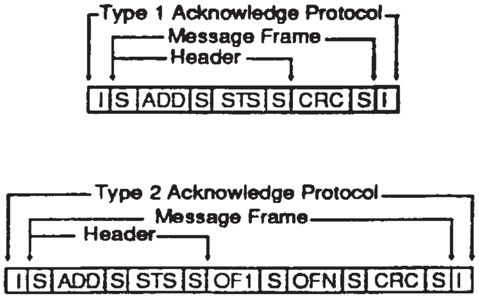
Type 1 Acknowledge Protocol
Message Frame
Header
I S ADD S STS S CRC S I
Type 2 Acknowledge Protocol
Message Frame
Header
I S ADD S STS S OF1 S OFN S CRC S I

FIGURE 7 Expanded Acknowledge Protocols

In Figure 7, examples of the two types of CASP acknowledge protocols are shown. Type 1 has a message frame containing the header's ADD and STS fields, and the CRC field. Type 2 has a message frame containing the header's ADD and STS fields, optional fields (OF) 1-N, and the CRC field.

In response to receiving either acknowledge protocol type (1 or 2) from the CASP, the PSBM checks that the correct address field was received, then checks the status field for errors. After checking the address and status fields, the PSBM receives the remainder of the acknowledge protocol. At the end of the acknowledge protocol, the PSBM matches the received CRC field against a CRC it calculates on the received data. If the correct address field was received, the status field indicates no errors, and the CRCs match, the PSBM is assured that the CASP has properly received and executed the command sent in the previous select protocol. If a failure occurred in the address or status field test, the PSBM knows that the CASP did not properly receive the previous command select protocol. If the address and status fields test passed, but a CRC error occurred, the PSBM knows that an error in the optional data fields following the header fields occurred. In response to an acknowledge protocol error, the PSBM can resend the command via another select protocol.

4.3 Pausing During a Protocol Transfer

During the transmission of a Type 2 select or acknowledge protocol it may be necessary to pause the transfer of fields within the message frame, due to memory limitations of the PSBM and/or CASP. For example, if a large number of optional fields is being sent, the transmitting or receiving device may not have sufficient memory to allow all the message frame fields to be transferred at one time. It is necessary, therefore, to provide a method of pausing the transfer of message frame fields so that the memories of the transmitter and receiver can be periodically downloaded from or uploaded to a larger memory, such as a disk drive.

A pausing capability can be easily realized by having the transmitting device (PSBM or CASP) output additional S signals following the S signal that separates the fields. Using this approach, pausing can occur between any two field frames. The length of the pause is determined by the number of additional S signals output from the transmitter. The transferring of fields within the message frame is resumed when the transmitting device outputs a D signal to start the next field.

4.4 CASP Command Set

The following commands form the basic CASP command set. Some of the commands support connecting the CASP's secondary port up to either the primary or remote port, while other commands support data transfer operations between a PSBM and a board resident memory, via the CASP's I/O port. Other commands can be added as required.

4.4.1 Connect PSBM Command

When the PSBM needs to access the board ICs, it sends a connect PSBM command to the CASP via a type 1 select protocol. In response to the connect PSBM command, the CASP sets the "PSBM connected" status bit, sends a reply to the PSBM using a type 1 acknowledge protocol, then connects its primary and secondary ports.

After receiving the acknowledge protocol from the CASP and verifying the "PSBM connected" status bit is set, the PSBM can access the board ICs using the 1149.1 bus protocol. After the connect PSBM command has been input to the CASP, other commands that do not effect the connection between the primary and secondary ports, such as the read and write commands, can be input to and executed by the CASP.

4.4.2 Disconnect PSBM Command

When the PSBM completes its access of the board ICs, it sends a disconnect PSBM command to the CASP via a type 1 select protocol. In response to the disconnect PSBM command, the CASP resets the "PSBM connected" status bit, sends a reply to the PSBM using a type 1 acknowledge protocol, then disconnects its primary and secondary ports. After receiving the acknowledge protocol from the CASP, the PSBM verifies the "PSBM disconnected" status bit has been reset.

4.4.3 Connect RSBM Command

When the PSBM requires the RSBM to access the board ICs, it sends a connect RSBM command to the CASP via a type 1 select protocol. In response to the connect RSBM command, the CASP sets the "RSBM connected" status bit, sends a reply to the PSBM using a type 1 acknowledge protocol, then connects its remote and secondary ports. After receiving the acknowledge protocol from the CASP, the PSBM verifies the "RSBM connected" status bit is set.

After the connect RSBM command has been input to the CASP, other commands that do not effect the connection between the remote and secondary ports, such as the read and write commands, can be input to and executed by the CASP. For example, the PSBM can send a command to the RSBM, via the CASP's I/O port, to initiate the remote access operation using a write command. Further, the PSBM can monitor the status of the remote access operation, via the CASP's I/O port, using a read command.

4.4.4 Disconnect RSBM Command

When the PSBM determines that the remote access of the board ICs is complete, it sends a disconnect RSBM command to the CASP via a type 1 select protocol. In response to the disconnect RSBM command, the CASP resets the "RSBM connected" status bit, sends a reply to the PSBM using a type 1 acknowledge protocol, then disconnects its remote and secondary ports. After receiving the acknowledge protocol from the CASP, the PSBM verifies the "RSBM Connected" status bit has been reset.

4.4.5 Write Command

When data is to be transferred from the PSBM to a memory via the CASP's I/O port, the CASP receives a write command from the PSBM via a type 2 select protocol. The write command select protocol message frame contains: a header with the CASP address and write command, a starting address field where the first data field will be written, a count field indicating the number of data fields to be written, one or more data fields, and a CRC field.

At the beginning of the write command select protocol, the CASP checks its address against the received address field and the write command against known commands. In response to the write command, the CASP outputs the received starting address field on the I/O port and stores the write count field. Next, the CASP writes the first received data field to the addressed memory location and decrements the count field. If the count field is not zero after the first write operation, the ASP increments the starting memory address, writes the next received data field, and decrements the count field again. These steps are repeated until the count field decrements to zero.

When the count field decrements to zero, the CASP realizes that the last data field has been received and written to memory and the next field received is the CRC. The CASP compares the received CRC with a CRC it has calculated on the received data to check for errors, then sends an appropriate reply to the PSBM using a type 1 acknowledge protocol. After receiving the acknowledge protocol from the CASP, the PSBM verifies that the write command was successful by checking the address and status fields within the header and the CRC field for errors.

While multiple data fields will usually be transferred during the write command to upload a test program or data, a single data field can be transmitted by simply setting the count field to one. An example of a single data write command is when the PSBM sends a command to the RSBM instructing it to execute a remote test operation, such as "initiate self-test #1".

4.4.6 Read Command

When data is to be transferred from a memory to the PSBM via the CASP I/O port, the CASP receives a read command from the PSBM via a type 2 select protocol. The read command select protocol message frame contains: a header with the CASP address and read command, a starting address field where the first data field will be read, a count field indicating the number of data fields to be read, and a CRC field.

At the beginning of the read command select protocol, the CASP checks its address against the received address field and the read command against known commands. In response to the read command, the CASP stores the received starting address and read count fields, then matches the received CRC against the calculated CRC.

After the read command select protocol completes, the CASP outputs the starting memory address from the I/O port, reads the first data field from memory, decrements the read count field, and starts a type 2 acknowledge protocol to transfer the data read from the memory to the PSBM. If the read count field is not zero after the first read operation, the CASP increments the memory address, and repeats the memory read and acknowledge protocol transfer sequence. When the read count field decrements to zero, the CASP realizes that the last data field has been read from memory. After sending the last data field to the PSBM, the CASP sends the calculated CRC field then terminates the acknowledge protocol.

In response to the read command acknowledge protocol, the PSBM checks the header's address and status fields, receives and stores a predetermined number of data fields, then matches the received CRC field against a calculated CRC. If no errors are found, the PSBM is assured that the read command has been executed and the data received is correct.

While multiple data fields will usually be transferred during the read command acknowledge protocol, to download test or system data, a single data field can be transmitted by simply setting the count field to one. An example of a single data read command is when the PSBM needs only to read the status of the RSBM.

4.4.7 Read Status Command

To allow the PSBM to read the CASP's internal status register, the PSBM sends a read status command to the CASP via a type 1 select protocol. The read status command select protocol message frame contains a header with the CASP address and read status command, and a CRC field. After checking the read status command select protocol's address, command, and CRC fields, the CASP sends a type 1 acknowledge protocol to the PSBM to transfer its status register field to the PSBM. In response to the read status command acknowledge protocol, the PSBM can check the settings of the CASP's status register bits.

The read status command, like the read and write commands, can be executed without effecting any of the connection type commands. The read status command allows the PSBM to monitor the internal status of CASP, as well as external status inputs, such as the RSBM's interrupt input. The following list defines the required status bits in the CASP's status register.

Command Error – A status register bit indicating an unknown command was received.

CRC Error – A status register bit indicating a mismatch between the received and calculated CRC.

Interrupt Request – A status register bit indicating the occurrence of an external interrupt request.

Primary Port Connected – A status register bit indicating a connection between the primary and secondary ports.

Remote Port Connected – A status register indicating a connection between the remote and secondary ports.

Secondary Port Idle – A status register bit indicating that the secondary port is idle (STMS=0).

Secondary Port Reset – A status register bit indicating that the secondary port is reset (STMS=1).

4.5 Global Commanding

To support a method of executing global commands, CASPs can include a global command address (GCA), similar to the reset address (RSTA) in Figure 4. During a global command select protocol, all CASPs respond to the GCA to execute the command that follows. To avoid bus contention on PTDO, CASPs do not output acknowledge protocols in response to global command select protocols.

5.0 Adapting the CASP for 2-wire Backplanes

As mentioned in the "Interfacing P1394 to 1149.1" section, newer 32-bit IEEE backplane standards only allocate two wires for a backplane level serial bus. If this trend continues, only a 2-wire serial bus, such as P1394, can be used in future backplanes, since both 1149.1 and P1149.5 require more than two bus wires. While P1394 is a good high speed 2-wire communications bus, its complex protocols and circuitry may be too sophisticated and/or costly for the simpler access applications involved in serial testing. To provide an alternate 2-wire serial bus for newer backplane standards, the primary port of the CASP can be easily adapted to communicate the select and acknowledge protocols using only two wires.

In Figure 8, the CASP's primary port is shown interfaced to a 2-wire PSBM via a serial input/output (SIO) wire and a clock (CLK) wire. The SIO wire combines the unidirectional TDO and TDI backplane wiring channels into a single bidirectional wiring channel. The SIO wire is capable of transferring the select and acknowledge protocols between the PSBM and CASP in response to the CLK output from the PSBM. The select and acknowledge protocols can both be transmitted on a single wire since they are transmitted at different times, as shown in Figure 3. When the CASP is used in a 2-wire backplane, the connect and disconnect PSBM commands are no longer valid, since an 1149.1 connection is not possible in two wires. However, all the other CASP commands can be used, as previously described, to enable remote test access and data transfer operations.

Using this approach, a cost-effective 2-wire serial bus can optionally be used in newer backplane standards, in place of P1394, to provide a simple access method for test and data transfer operations. An important advantage of this approach is that it allows a common backplane test methodology to be developed and practiced, independent of the physical interfaces used in various system backplanes. For example, a backplane test access program, designed using a common CASP command set, could be used in either a 4 or 2-wire serial backplane environment. The combination of a common set of backplane test commands and a flexible serial interface, provides the building blocks from which a standard backplane test access language could be developed.

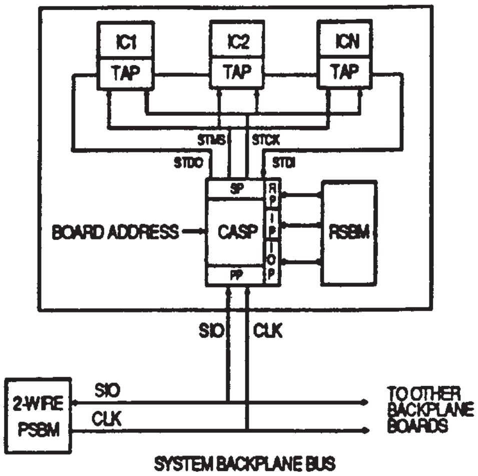

FIGURE 8 2-Wire CASP Backplane Interface

In future updates to the current 32-bit backplane buses, and in definitions of future 64-bit buses, it may be beneficial for test representatives to participate in backplane working groups to help specify backplane test bus requirements. Since a need will probably exist for both a 2-wire communications bus (like P1394) and a 2-wire test bus (like the one proposed here), an ideal scenario would be for backplane standards to specify two pairs of backplane wires so that both serial bus types could be supported. In this way, both serial buses could operate full-time in a backplane environment to optimally perform the task each was designed for. Also, since separate interfaces for the higher performance communications bus and lower performance test bus could be designed more efficiently, a lower cost of implementation would probably be achieved. Further, if separate serial buses existed at the backplane level for test and communication, the software for each bus could be designed more efficiently and executed in parallel.

6.0 Summary and Conclusion

This paper has described a new approach of accessing 1149.1 based boards in a backplane environment. The approach can be used as a simple backplane to board connection method or expanded to include commandable features considered useful in a backplane environment.

The ASP provides a simple backplane to board connection circuit to effectuate expanded use of the 1149.1 test bus. The select and acknowledge protocols can be transmitted between a PSBM and ASP using normal 1149.1 scan operations, assuming the PSBM can hold its TMS signal at a 1 or 0 during scan. Therefore, existing 1149.1 bus masters and software can easily be adapted to use this approach. The ASP circuit is simple and can be assembled in small footprint packages (20 pins), resulting in a low cost, low area overhead, backplane-to-board level 1149.1 test interface.

The commandable ASP (CASP) provides a way of expanding the basic ASP and protocols to suit the needs of more demanding applications, while maintaining protocol compatibility with the 1149.1 standard. The ability of the CASP to support board resident RSBMs, provides a structured method of designing systems capable of distributed test control. The ability of the CASP to support data transfer, provides a method of emulating the data transfer features being developed in the IEEE P1149.5 and P1394 serial bus standards. The ability of the CASP to operate in a 2-wire backplane environment, allows it to serve as an alternate serial bus to P1394 when only simple test access is required in a system.

The advantages offered by this new approach are: (1) direct compatibility with 1149.1, (2) improved serial data transfer bandwidth, (3) simple, cost effective implementation, (4) easily expandable to include other features useful in backplane applications, and (5) capable of being used in either a 4 or 2-wire serial bus backplane environment.

References

1 *IEEE Std 1149.1-1990, Standard Test Access Port and Boundary Scan Architecture*
2 *IEEE P1149.5, Standard Module Test and Maintenance Bus Protocol*
3 *J. Andrews, "IEEE's 1149.5 Bus Facilitates JTAG", ATE & Instrumentation Conference, January 1991.*
4 *IEEE P1394, High Speed Serial Bus*
5 *IEEE/ANSI 896, Futurebus+*
6 *IEEE/ANSI 960, FASTBUS*
7 *IEEE/ANSI 1014, VMEbus*
8 *IEEE/ANSI 1295, Multibus II*
9 *D. Bhavsar, "An Architecture for Extending the IEEE Standard 1149.1 Test Access Port to System Backplanes", International Test Conference, October 1991.*