Application Report

SNLA055H – March 2003 – Revised April 2013

AN-1259 SCANSTA112 Designer's Reference

ABSTRACT

This application note primarily applies to the SCANSTA112. The feature set of the SCANSTA112 is a superset of that of the SCANSTA111. Much of the content of this application note applies to both devices. With its richer feature set, the SCANSTA112 provides significant additional capability to the system designer. Using this capability is the subject of this application note.

Contents

List of Figures

List of Tables

1 Introduction

In 1990 the IEEE approved a draft standard specifying a test access port for integrated circuits. The initial work for this standard had been performed by a working group made up of primarily European and North American companies interested in addressing printed circuit (PC) board test problems. This working group was called the Joint Test Action Group, or JTAG.

The draft standard was motivated by the fact that the problem of testing PC boards had become, over time, increasingly intractable. Component densities on PC boards had increased. Making connections to integrated circuits on such boards for test purposes had become more difficult. The draft standard was designed to define a standard technique for improving the testability of PC boards.

The draft standard that grew out of the work of JTAG was *IEEE Standard 1149.1-1990*, “IEEE Standard Test Access Port and Boundary Scan Architecture”, sometimes called IEEE 1149.1, boundary scan, or JTAG. The current version of this standard as of this writing is *IEEE Standard 1149.1-2001*.

The standard specifies the structure and function of a secondary I/O port used to access the pins of an integrated circuit for testing purposes. This secondary I/O port is called the Test Access Port or TAP. It is often referred to as the IEEE 1149.1 TAP, the JTAG TAP, or the boundary scan TAP.

Access to the pins of the integrated circuits on a PC board affords the system designer a powerful capability to test the integrity of connections on the board or between boards. It also enables in-system programming of programmable devices on the board. The JTAG TAP provides an interface to programmable devices through which they can be reprogrammed.

In its most basic form the IEEE 1149.1 standard envisions a single, one-bit serial link connecting all the boundary scan-enabled devices on a PC board. The data output of one device is connected to the data input of the next. The clock and control lines are common to all devices. All boundary scan-enabled devices are designed to support this basic architecture. This is the simplest way to connect a one-bit interface from device to device and for small, simple PC boards it is an appropriate architecture.

For PC boards with many devices this simple architecture can become unwieldy. It is often desirable to address a limited set of the devices on a PC board through the JTAG TAP. Sometimes it is necessary to address only a single device. This is frequently the case for in-system programming of programmable logic devices. Addressing only a subset of the boundary scan-enabled devices on a PC board is not possible with the most basic IEEE 1149.1 boundary scan architecture. A support device for the IEEE 1149.1 test access port that enables such addressing is required.

Texas Instruments™ manufactures two devices to address this requirement: the SCANSTA111 and the SCANSTA112. Both devices enable partitioning and multiplexing of IEEE 1149.1 boundary scan chains. The SCANSTA112 7-Port Multidrop IEEE 1149.1 (JTAG) Multiplexer provides seven local scan ports. The SCANSTA111 Multidrop Addressable IEEE 1149.1 Port provides three local scan ports. Operation of the two devices is similar.

Both of these devices partition a single scan chain into multiple local scan chains. The devices are addressable so that more than one JTAG multiplexer can be used with a single master scan chain. Multiple PC boards, each with an IEEE 1149.1 TAP and with one or more SCANSTA111s or SCANSTA112s on board, can utilize a single backplane test bus for system-level IEEE 1149.1 TAP access. This simplifies the design of a system to implement boundary scan.

Before we describe the application of the SCANSTA112 in a boundary scan system, we will describe the basic operation of a single boundary scan chain. This will provide background for the application of the SCANSTA112.

2 Boundary Scan Chain Configurations

A boundary scan-enabled system may be designed with a single boundary scan chain or with multiple boundary scan chains. A design with multiple boundary scan chains requires a JTAG multiplexer such as the SCANSTA112. The architectures of both the single and multiple boundary scan chain systems are described in this section.

2.1 Single Boundary Scan Chain Configuration

When the IEEE 1149.1 boundary scan chain is not partitioned by a scan chain multiplexer, the connections for the test access port (TAP) are made as shown in Figure 1. As the figure indicates, there are five signals associated with the TAP. These are Test Clock TCK, Test Mode Select TMS, Test Data In TDI, Test Data Out TDO, and Test Reset $\overline{\text{TRST}}$. The $\overline{\text{TRST}}$ signal is optional according to the IEEE 1149.1 standard.

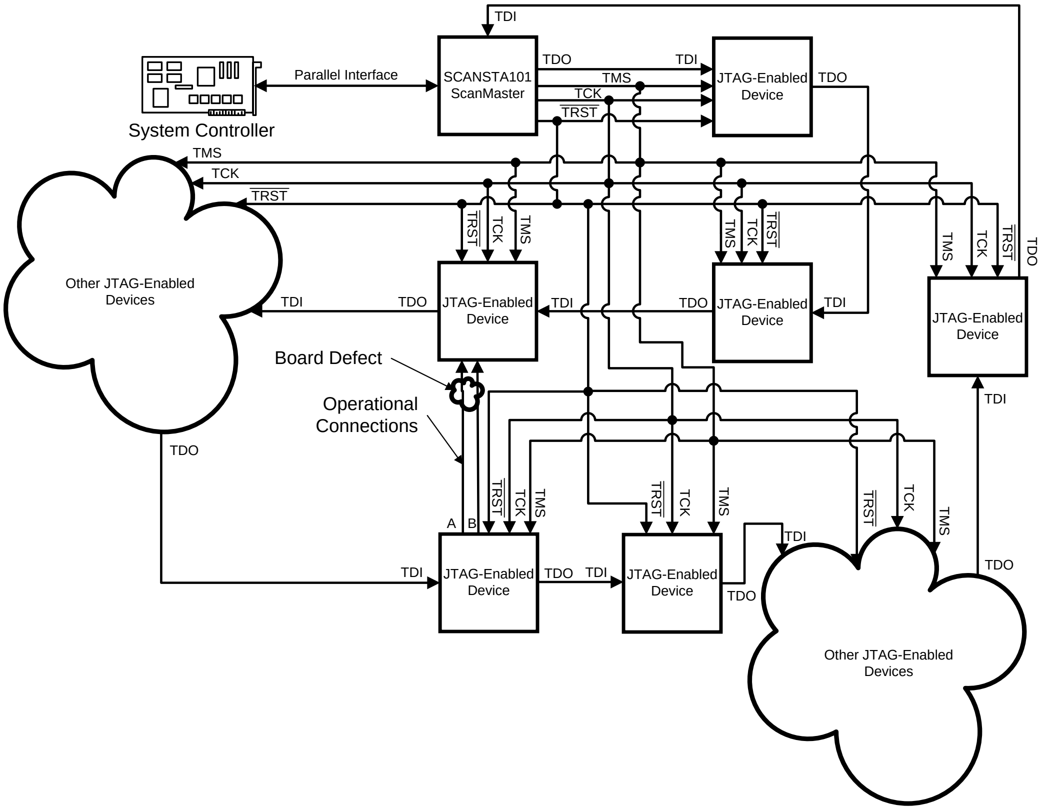

Figure 1. Example of a Simple Boundary Scan Chain

A boundary scan controller provides the TCK signal. It also drives the TMS signal, the TDO signal, and the $\overline{\text{TRST}}$ signal. The boundary scan chain output returns to the boundary scan controller through the TDI signal. The term boundary scan controller is used in this document to refer to the master boundary scan device that controls all boundary scan operations over the IEEE 1149.1 TAP. In Figure 1 the boundary scan controller is a TI™ SCANSTA101 ScanMaster.

Each boundary scan-enabled device includes a state machine called the TAP controller. The sequence of states through which this state machine can be transitioned is shown in Figure 2. The configuration and operation of the TAP is determined by the state of the TAP controller. By sequencing the TAP controller through a set of desired states the boundary scan controller causes the system to perform the desired operations using the IEEE 1149.1 TAP.

In Figure 2 the names of the states are abbreviated. *Data-Register* is abbreviated *DR* and *Instruction Register* is abbreviated *IR*. Thus the state name referred to as *Shift-Data-Register* in the text is shown as *Shift-DR* in Figure 2.

Referring again to Figure 1, note that the TCK, TMS, and $\overline{\text{TRST}}$ signals are routed to all the boundary scan-enabled devices in parallel. When the asynchronous reset $\overline{\text{TRST}}$ signal is asserted the TAP controllers of all the boundary scan-enabled devices go to the *Test-Logic-Reset* state at the same time. Each of the boundary scan-enabled devices receives TCK transitions at the same time. Each device is required to write data to its TDO output on the falling edge of TCK when it is in the *Shift-Instruction-Register* or *Shift-Data-Register* state. Each device detects the state of the TMS input and the TDI input on the rising edge of TCK. Since all the devices are clocked by the same signal they all remain in synchronous operation when they are in boundary scan test mode.

Since all the boundary scan-enabled devices receive the same TMS signals, all their TAP controllers transition from one state to the next as shown in Figure 2 in lockstep. The entire boundary scan chain has a single TAP controller state. The TAP controller is reset for the entire boundary scan chain by asserting $\overline{\text{TRST}}$ or by holding the TMS signal high for five successive TCK pulses (this is called a five-high TMS reset). Following the reset, the TAP controller is in the *Test-Logic-Reset* state. The boundary scan controller drives the value on the TMS line to sequence the TAP controller through its various states.

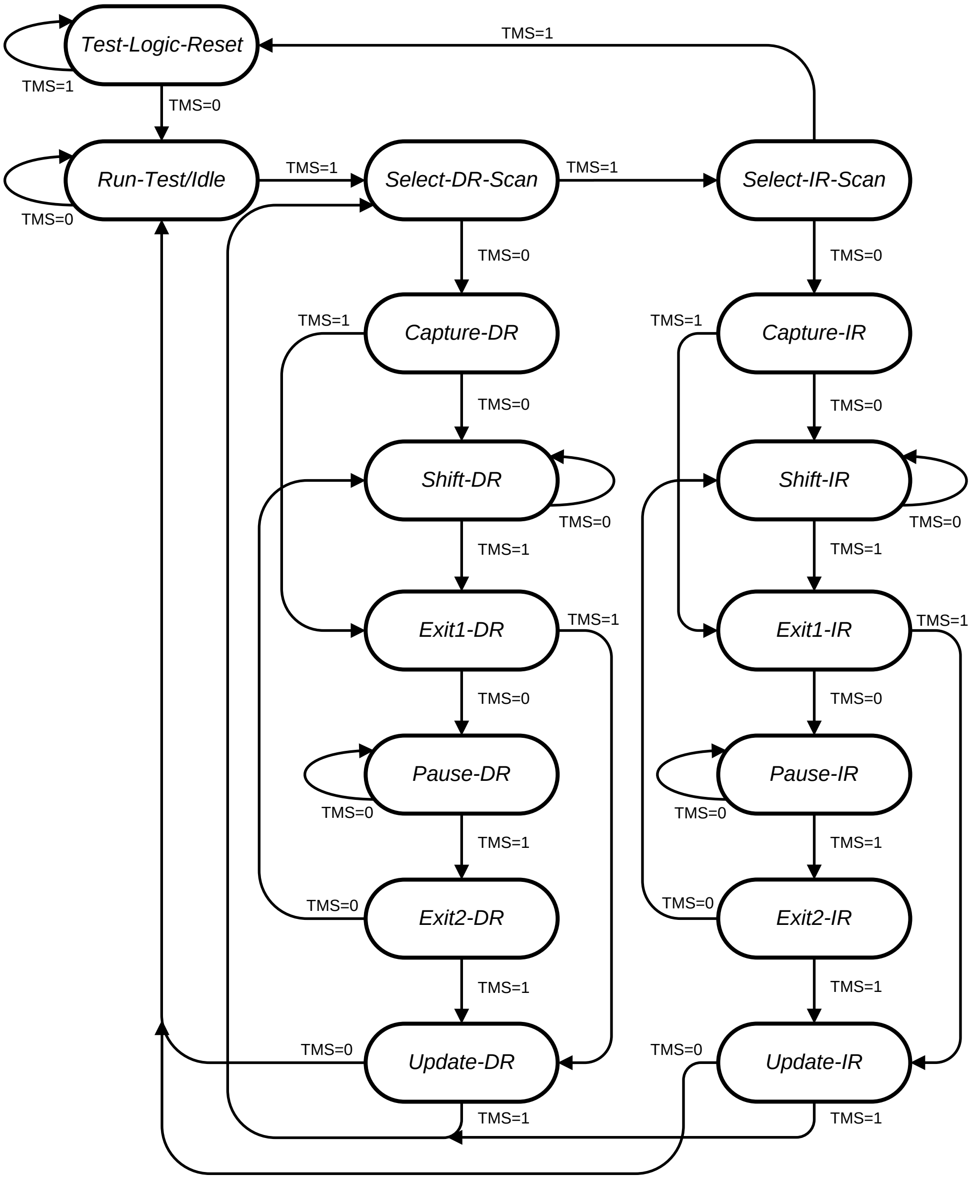

Figure 2. TAP Controller State Machine

As noted, the TAP controllers of all the devices in the scan chain go through the same sequence of states. For example, starting from the *Test-Logic-Reset* state, when the boundary scan controller drives on the TMS line a bit sequence of 01100 on five successive TCK pulses the TAP controller is sequenced to the *Shift-Instruction-Register* state.

As indicated in Figure 1 the TDI line of the first device in the scan chain is connected to the TDO line from the boundary scan controller. The TDO line of this first device is connected to the TDI line of the next device. This continues until the last device in the scan chain has its TDO line connected to the TDI line of the boundary scan controller. The boundary scan controller presents bits to the TDI pin of the first device. They are shifted through a register inserted in the scan chain by the first device until the register is full, at which point the bits are shifted out of the TDO pin of this device to the TDI pin of the next one.

As an example, consider the case where the TAP controller is in the *Shift-Instruction-Register* state. Then each boundary scan-enabled device inserts its instruction register into the scan chain. The bits received at the TDI input of the first device are shifted through the instruction registers of each device in the chain at each TCK pulse as long as the TMS line is driven with a 0. When the boundary scan controller eventually drives the TMS line with a 1, the TAP controller enters the *Exit1-Instruction-Register* state. If the boundary scan controller continues to drive a 1 on the TMS line, the TAP controller enters the *Update-Instruction-Register* state. In this state the instructions loaded into the instruction register become active. The boundary scan-enabled devices respond to these instructions in device-specific ways to perform the desired boundary scan operations.

This is the architecture of a basic system with a single boundary scan chain. This architecture is fine for systems with only a few boundary scan-enabled devices with simple connections between them. It is less than optimal for systems with many boundary scan-enabled devices. It is especially cumbersome when some devices must be accessed extensively and repeatedly through the TAP. A programmable logic device, for example, will often require execution of many boundary scan operations in order to program it. Programming a programmable logic device that is part of a long boundary scan chain requires shifting many additional bits into the scan chain beyond just those required to program the device. In some cases, the software tools used to generate the programming sequences for these devices cannot automatically generate sequences including these additional required bits. When the software tools fail to automate the process of producing the required sequences, manual changes to the programming files are required. This is time-consuming and error-prone.

A better approach to the boundary scan architecture in a complex system is a partitioned scan chain. The SCANSTA112 makes it possible to partition a single long scan chain into multiple shorter local scan chains. We will describe the architecture of such a partitioned boundary scan system next.

2.2 Partitioned Boundary Scan Chain Configuration

As noted, in systems with many boundary scan-enabled devices, and especially in systems with programmable devices, partitioning the scan chain can simplify the development of boundary scan test sequences and improve the throughput of boundary scan operations. The SCANSTA112 JTAG multiplexer can be used to partition the scan chain. The SCANSTA112 provides seven local scan ports (LSPs) to which local scan chains may be attached.

An example system using the SCANSTA112 JTAG Multiplexer for scan chain partitioning is shown in Figure 3. As the figure indicates, the backplane TAP lines of the SCANSTA112 (TCK_B, TMS_B, TDO_B, TDI_B, and $\overline{TRST_B}$) are connected to the master boundary scan TAP in the system. The boundary scan controller controls this TAP. The boundary scan controller can send instructions through the master TAP to the SCANSTA112. These instructions can be used to address and configure the SCANSTA112; to select which local scan ports (LSPs) are to be active; and to perform boundary scan operations using both the SCANSTA112 and the scan chains attached to its LSPs.

The SCANSTA112 can also operate in a pin-controlled mode as opposed to being controlled over the IEEE 1149.1 TAP as described above. Control of the SCANSTA112 in pin-controlled mode is accomplished by the SB/$\overline{S}$ pin, the $LSPsel_{(0–6)}$ pins, and the TRANS pin. Both the IEEE 1149.1 TAP-controlled and pin-controlled modes of operation are described in this application note.

In general, the primary mode of operation where the SCANSTA112 is controlled by the backplane TAP is called ScanBridge mode. The mode where the SCANSTA112 is controlled by the device pins is called stitcher mode. This represents an oversimplification of the description of the primary operating modes since there is considerable overlap between them. A more complete description is presented in Section 3.1.

There are multiple scan chains in the system shown in Figure 3. Each is attached to a local scan port (LSP) of the SCANSTA112. The system supports various boundary scan operations. For example, a programmable logic device can be programmed using its LSP while the TAP controllers of all the other scan chains are held in *Test-Logic-Reset* state. Boundary scan operations on a scan chain on one LSP may be conducted while the TAP controllers of all the scan chains on other LSPs remain in *Test-Logic-Reset* state. The TAP controllers on one or more LSPs can also be held in other stable states such as *Run-Test-Idle* while boundary scan operations are performed on other scan chains. Multiple local scan chains can be chained together through the multiplexer. A highly-configurable boundary scan chain can be implemented using the SCANSTA112.

As Figure 3 indicates, each LSP is equipped with all five boundary scan TAP signals. For each LSP, the TCK, TMS, and $\overline{\text{TRST}}$ lines are connected to all the devices in the corresponding local scan chain as if the local scan chain were the only scan chain present in the system. The TDO line of the LSP is connected to the TDI line of the first device in the local scan chain. In each local scan chain the TDO signal from one device is connected to the TDI signal of the next just as in the basic boundary scan chain in Figure 1. The TDO line of the device at the end of the local scan chain is attached to the TDI line of the LSP.

The SCANSTA112 is normally configured so that it appears to the boundary scan controller to be part of the selected local scan chain. It can also be configured as a transparent multiplexer in what is referred to as transparent scan chain mode. In transparent scan chain mode, the boundary scan controller is not aware that the SCANSTA112 is in the system. All the configuration modes available for the SCANSTA112 will be discussed in this application note.

It is possible to configure a system with multiple SCANSTA112 JTAG Multiplexers, each providing multiple LSPs. Such a system can be either flat or hierarchical.

Each SCANSTA112 in a flat system is assigned a unique address using the address pins on the device $S_{(0-7)}$. To initiate boundary scan operations in a system with multiple SCANSTA112s the boundary scan controller first addresses one or more SCANSTA112s using their assigned addresses. Once the desired SCANSTA112 or group of SCANSTA112s has been addressed the boundary scan controller configures their LSPs. The boundary scan controller then initiates boundary scan operations on the selected LSPs.

A hierarchical system can provide further partitioning of the boundary scan chain. The backplane TAP of one SCANSTA112 is driven from one of the LSPs of another SCANSTA112. Such a hierarchical system provides expanded flexibility. A hierarchical system also provides an expanded address space for the SCANSTA112s. It is only required that SCANSTA112s at lower levels in the hierarchy have addresses unique within the group of SCANSTA112s attached to the same LSP.

The SCANSTA112 is designed to support multiple levels of hierarchical connections. Implementation of such a hierarchical system is a natural extension of the implementation of a flat system with multiple SCANSTA112s.

The first step in performing boundary scan operations on a system using the SCANSTA112 to partition the scan chain is to select the SCANSTA112 or group of SCANSTA112s to use for the boundary scan operations. The sequence of operations for selecting a SCANSTA112 or a group of SCANSTA112s depends upon the design of the system. In particular, it depends upon the primary operating mode of the SCANSTA112. We next describe the various scenarios for the selection of SCANSTA112s in a system.

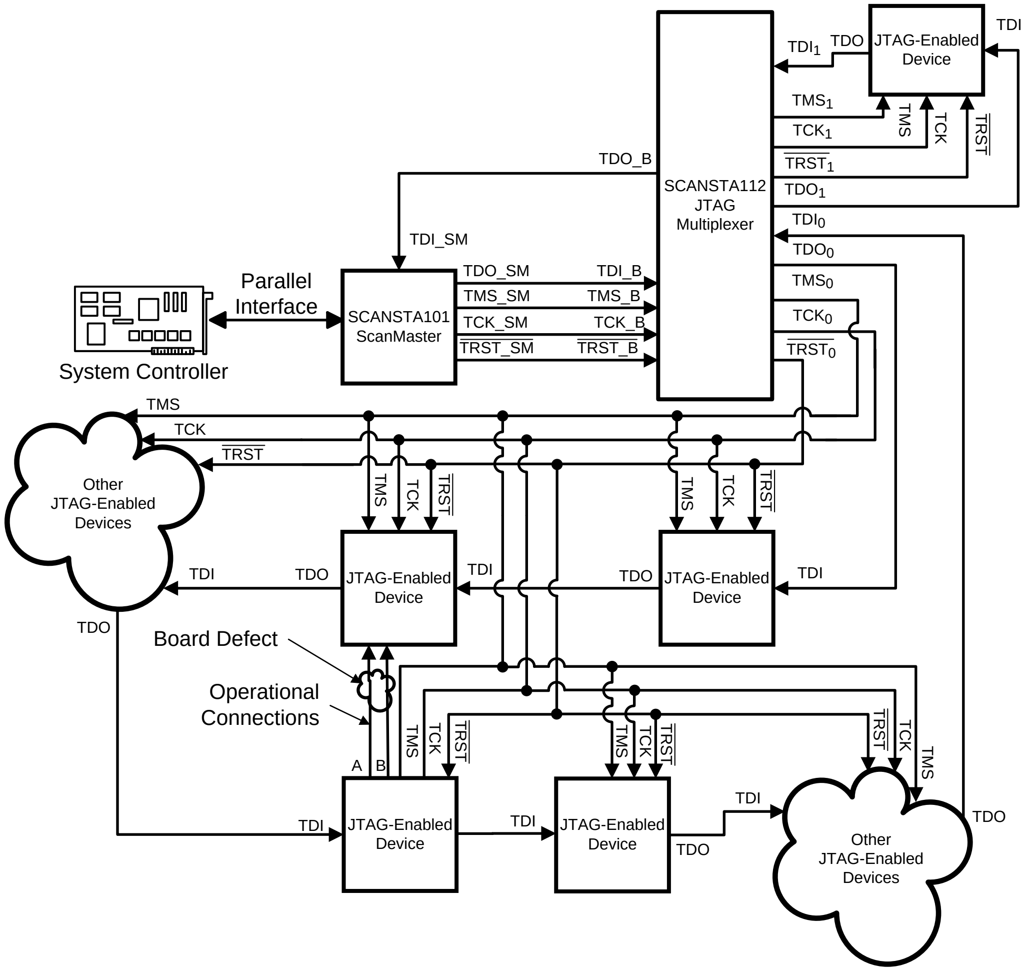

Figure 3. Example of a Partitioned Boundary Scan Chain

3 SCANSTA112 Selection and Addressing

There are several possible scenarios for the selection of SCANSTA112s in a system. To determine which scenarios are to be used in a given system we consider the number of SCANSTA112s in the system, how they are arranged, and how they are to be used.

If there is more than one SCANSTA112 in a system, a user might wish to select a single SCANSTA112 and communicate only with devices accessible through that SCANSTA112. The user may wish to program a single programmable device on a given LSP of a given SCANSTA112; the user may wish to perform boundary scan operations on a single local scan chain on one of the SCANSTA112s; or the user may wish to perform boundary scan operations on multiple local scan chains all connected to one of the SCANSTA112s. In any of these cases, the user must first select the correct SCANSTA112 before boundary scan operations can begin.

Alternatively, the user might wish to select all the SCANSTA112s in the system at once. Consider the case where a system consists of a number of identical printed circuit boards on a backplane, each equipped with a SCANSTA112. Suppose that the same boundary scan operation is to be performed on all the boards in the system simultaneously: a board test, programming of a programmable device, or some other boundary scan operation. In this case the user will wish to select all the SCANSTA112s in the system simultaneously and allow them all to perform the required operation on their respective boards. Note that even though all the SCANSTA112s in the system are to be selected at once, they cannot all be allowed to drive the backplane TDI line, to which they are all connected, at once. The SCANSTA112 features a special addressing mode to permit addressing all the SCANSTA112s in a system while preventing contention on the backplane TDI line.

Finally, the user might wish to select some, but not all, of the SCANSTA112s in the system and perform boundary scan operations only through the selected devices. A system with some PC boards of one type and some of another type on the same backplane, each equipped with a SCANSTA112, might call for this selection scenario. In this case the user might select the SCANSTA112s on all PC boards of one type for boundary scan operations. The unselected SCANSTA112s would not participate in these boundary scan operations. Note that this mode of operation also requires special consideration to prevent contention between multiple drivers on the backplane TDI line. The SCANSTA112 has a special addressing mode to accommodate this type of operation while preventing contention on the backplane TDI line.

The two primary operating modes of the SCANSTA112 are called ScanBridge mode and stitcher mode. The selection scenarios that require selection of a SCANSTA112 or a group of SCANSTA112s from among multiple SCANSTA112s in a system require the use of ScanBridge primary operating mode. In the next section we describe both primary operating modes.

3.1 SCANSTA112 Primary Operating Modes

The SCANSTA112 features two primary operating modes, ScanBridge mode and stitcher mode. We will postpone for the moment describing how the primary operating mode is selected and focus first on the differences in operation of the SCANSTA112 when it is in each of these primary operating modes.

The most important difference between these modes is that when a SCANSTA112 is in stitcher mode, it is always selected and always in its operational mode.

Consider a system in which there is only one SCANSTA112. In such a system it is not necessary to select a single SCANSTA112 from a set of SCANSTA112s because, obviously, there is only one SCANSTA112 to select. A SCANSTA112 in stitcher mode is always selected. This may be the desired behavior in a system with a single SCANSTA112.

Now compare such a system, with a single SCANSTA112, to a system with multiple SCANSTA112s. If all the SCANSTA112s in such a system were in stitcher primary operating mode, and were thus always selected, multiple SCANSTA112s might try to drive the backplane TDI line simultaneously, resulting in bus contention. Such a system will not work properly if the SCANSTA112s are in stitcher mode.

In a system with a single SCANSTA112 there is only one SCANSTA112 to drive the backplane TDI line. In such a system the SCANSTA112 may be used in stitcher primary operating mode, where it is always selected. A system like this, with a single SCANSTA112, really represents the only situation where the stitcher primary operating mode should be used.

Even when there is only one SCANSTA112 in a system, ScanBridge mode may still be used. So why would a system designer ever use stitcher mode?

Stitcher mode can simplify boundary scan operations by eliminating the need to configure the SCANSTA112 (here we assume that there is only one SCANSTA112 in the system) over the IEEE 1149.1 TAP. In stitcher mode the default configuration of the SCANSTA112, including which LSPs are selected and whether or not the SCANSTA112 is in transparent mode, is determined by logic values at the pins of the device. These logic values can be hardwired or can be supplied by any logic device in the system. This means that in stitcher mode the operation of the SCANSTA112 may be, though it need not be, entirely controlled by logic values on the SCANSTA112's pins. This can simplify operations over the IEEE 1149.1 TAP.

When the SCANSTA112 is operated in stitcher mode it is not necessary to send any commands to the SCANSTA112 to configure it. All the required configuration can be done by setting the appropriate logic values on the SCANSTA112's pins.

ScanBridge mode may be used whether there is a single SCANSTA112 in a system or more than one. When the SCANSTA112 operates in ScanBridge mode it is not always selected the way it is in stitcher mode. In fact, when the SCANSTA112s in a system are operated in ScanBridge mode all boundary scan operations must begin by selecting the SCANSTA112s which will participate in them — even if there is only one SCANSTA112 in the system. After the desired SCANSTA112s are selected, the desired LSPs must be configured and unparked. The additional configuration steps required when the SCANSTA112s are operated in ScanBridge mode increases the complexity of the required boundary scan sequences; but this increased complexity also provides increased flexibility which is often necessary for efficient boundary scan operation.

3.2 Choosing the SCANSTA112 Operating Mode

The criteria for selecting the SCANSTA112 primary operating mode are simple. If there is more than one SCANSTA112 in a system, use ScanBridge mode. If there is only one SCANSTA112 in a system, either ScanBridge or stitcher mode may be used.

Stitcher mode permits the user to do everything that can be done in a system with a single SCANSTA112. Since the SCANSTA112 may be controlled entirely from dedicated control pins in stitcher mode this may simplify boundary scan operations.

In ScanBridge mode the SCANSTA112 can be addressed, so it can also be deselected. This makes ScanBridge mode suitable for use in applications where there is more than one SCANSTA112 in the system.

All the device functionality available in stitcher mode is also available in ScanBridge mode. In ScanBridge mode the SCANSTA112 must be configured by instructions delivered via the backplane TAP. Additional functionality which is not available in stitcher mode is available in ScanBridge mode.

The requirements for configuring and operating the SCANSTA112 in each of its two primary modes will be discussed next. We begin with how to set the primary operating mode.

3.3 Setting the Primary Operating Mode

The primary operating mode, either ScanBridge or stitcher, is ultimately controlled by the value of bit 2 in the control register. The registers of the SCANSTA112 will be described in a later section of this application note. For now it is sufficient to note that the control register is one of the optional registers permitted by the IEEE 1149.1 standard. It is an 8-bit read/write register, the contents of which control several important operating characteristics of the SCANSTA112.

If bit 2 of the control register is a 1, the SCANSTA112 operates in ScanBridge mode. If it is a 0, the SCANSTA112 operates in stitcher mode. The default value of bit 2 in the control register is set to the value present on the SB/$\overline{\text{S}}$ pin when the SCANSTA112 is powered up. The value on the SB/$\overline{\text{S}}$ pin will set the primary operating mode of the SCANSTA112 when it is powered up. For example, if the SB/$\overline{\text{S}}$ pin on a SCANSTA112 is tied to a static 1 (high) then the SCANSTA112 will always enter ScanBridge mode immediately upon power up.

After the SCANSTA112 has been powered up in stitcher mode there are two ways to set the SCANSTA112 to ScanBridge mode without cycling power to it.

1. Drive the SB/$\overline{\text{S}}$ pin to a logic 1 (high) and reset the SCANSTA112. The reset can be accomplished by using the $\overline{\text{RESET}}$ pin, by using the $\overline{\text{TRST}}$ pin on the master backplane TAP, or by issuing a five-high TMS reset. The IgnoreReset bit in the control register, bit 0, when set to 1 will cause the SCANSTA112 to ignore the latter two reset methods listed above. The control register will be described later.
2. Write a 1 to bit 2 of the control register, the ScanBridge/Stitcher mode bit. This is not possible when the SCANSTA112 is in full transparent scan chain mode. As will be discussed below, the control register is not accessible in full transparent scan chain mode.

If the SB/$\overline{\text{S}}$ pin on the SCANSTA112 is tied to a static 0 (low) then the SCANSTA112 will always enter stitcher mode immediately upon power up. The SB/$\overline{\text{S}}$ pin should only be tied low in systems containing a single SCANSTA112.

After the SCANSTA112 has been powered up in ScanBridge mode there are two ways to set the SCANSTA112 to stitcher mode without cycling power to it.

1. Drive the SB/$\overline{\text{S}}$ pin to a logic 0 (low) and reset the SCANSTA112. The reset can be accomplished by

using the $\overline{\text{RESET}}$ pin, by using the $\overline{\text{TRST}}$ pin on the master backplane TAP, or by issuing a five-high TMS reset. The IgnoreReset bit in the control register, bit 0, when set to 1 will cause the SCANSTA112 to ignore the latter two reset methods listed above. The control register will be described later.

2. Write a 0 to bit 2 of the control register, the ScanBridge/Stitcher mode bit. This is not possible when the SCANSTA112 is in full transparent scan chain mode. As will be discussed below, the control register is not accessible in full transparent scan chain mode.

We will next describe the method for addressing one or more SCANSTA112s in a system. Note that this applies only to SCANSTA112s operating in ScanBridge mode. After the addressing operation in ScanBridge mode is described, we will describe the subsequent required operations in both primary operating modes.

3.4 Addressing Protocols in Scanbridge Mode

When the SCANSTA112 is operating as part of a scan chain defined by its LSPs it is said to be in operational state. In this state the SCANSTA112 is compliant to the IEEE 1149.1 standard except that the TDO output line may be at TRI-STATE (set to a high-impedance, non-driving state) even when the TAP controller is in the *Shift-Data-Register* or *Shift-Instruction-Register* states, as described below.

In ScanBridge mode the SCANSTA112 supports a two-level configuration protocol in addition to its operational state. In other words, in ScanBridge mode the SCANSTA112 sometimes behaves in a way that is not compliant to the IEEE 1149.1 standard. This behavior is necessary to permit the SCANSTA112 to perform its designed function of multiplexing boundary scan TAP signals. This behavior is referred to as the Level 1 and Level 2 protocols.

Level 1 protocol describes the required actions to address and select the desired SCANSTA112 when it is in ScanBridge mode.

Level 2 protocol applies after the SCANSTA112 has been addressed using the Level 1 protocol. Level 2 protocol describes the instructions delivered over the backplane TAP which are required to configure the multiplexing function and enable the connection (UNPARK) between the local scan chain and the backplane bus via a LSP.

3.4.1 Level 1 Protocol — Addressing the SCANSTA112

In a system with multiple SCANSTA112s, all the SCANSTA112s must operate in ScanBridge mode.

Consider the scenario where the user wishes to select only a single SCANSTA112 from multiple SCANSTA112s in a system. Each of the SCANSTA112s must be assigned a unique slot address. The boundary scan controller will begin the sequence of boundary scan operations by addressing the desired SCANSTA112 using its unique slot address.

The slot address of the SCANSTA112 is set by logic values on the $S_{(0–7)}$ slot address pins. The logic values on these pins form an 8-bit slot address. The usable address space is shown in Table 5.

Before the SCANSTA112 is addressed its TAP controller is not active. Its TAP is instead under the control of a higher-level state machine as described in Section 7.1.

The selection state machine starts out in the *Wait-for-Address* state. In this state the SCANSTA112 compares the value shifted into its instruction register with its slot address. When the SCANSTA112 detects that it has been addressed, it becomes available to receive configuration instructions over the backplane TAP.

A second scenario is one in which all the SCANSTA112s in the system are to be selected. In this scenario, instead of selecting a SCANSTA112 by using its unique slot address, all the SCANSTA112s in the system are selected simultaneously using the predefined broadcast address (see Table 5). The addressing operation is similar to that described above except that the address to be compared is the broadcast address and not the unique address of a single SCANSTA112.

When all the SCANSTA112s in the system are selected using the broadcast address, contention on the backplane TDI line is avoided by tri-stating (setting to a high-impedance, non-driving state) the TDO output drivers of all the SCANSTA112s. Boundary scan operations may be performed by scanning instructions into the instruction registers of all the SCANSTA112s and their associated LSPs. Results of boundary scan tests may be accumulated in each SCANSTA112 in its Linear Feedback Shift Register (LFSR). Reading results back from the SCANSTA112s will require that each SCANSTA112 be addressed singly, so a system that uses broadcast addressing must also provide unique slot addresses for each SCANSTA112.

A user may also wish to select some, but not all, of the SCANSTA112s in the system. In this scenario, instead of selecting a SCANSTA112 by using its unique slot address, a subset of the SCANSTA112s in the system are selected simultaneously using one of four predefined multi-cast group addresses. The addressing operation is similar to that described above except that the address to be compared is one of the four multi-cast addresses (see Table 5) rather than the unique address of a single SCANSTA112. The multi-cast address to which each SCANSTA112 will respond is determined by the contents of its Multi-Cast Group Register (MCGR). Setting the value of the Multi-Cast Group Register in each SCANSTA112 will require that each SCANSTA112 be addressed singly, so a system that uses multi-cast addressing must also provide unique addresses for each of the SCANSTA112s.

When a set of SCANSTA112s in the system are selected using a multi-cast group address, contention on the backplane TDI line is avoided as in broadcast address selection by tri-stating the TDO output drivers of all the selected SCANSTA112s. Boundary scan operations may be performed by scanning instructions into the instruction registers of each of the selected SCANSTA112s and their associated LSPs. Results of boundary scan tests may be accumulated in each selected SCANSTA112 in its LFSR as in the broadcast addressing scenario described above. Reading back the results from each selected SCANSTA112 also requires that each SCANSTA112 be addressed individually as in the broadcast addressing scenario.

Once the Level 1 protocol has been carried out and the desired SCANSTA112(s) have been selected, the desired LSPs for each SCANSTA112 must be selected and unparked. This is accomplished by the Level 2 protocol described next.

3.4.2 Level 2 Protocol — Selecting and Unparking the Local Scan Chains

After a SCANSTA112 operating in ScanBridge mode has been addressed its LSPs must be selected and unparked before boundary scan operations on those LSPs can begin. The contents of the mode registers (mode register 0, mode register 1, and mode register 2) of a SCANSTA112 operating in ScanBridge mode control the selection of LSPs. A detailed description of the mode registers appears in Section 8.1. For now, however, we will focus on the operations required on the backplane TAP to select and configure the desired LSPs.

Upon power up or reset the contents of the mode registers are reset to their default values. See Table 4 for the default register values. The default register values insert LSP_0 into the scan chain.

Immediately after the SCANSTA112 is addressed all the LSPs, including the selected LSP_0, will be parked in the *Test-Logic-Reset* state. Prior to beginning boundary scan operations the boundary scan controller must select any additional desired LSPs (and deselect LSP_0 if it is not to be used) and unpark the selected local scan chains.

Selection of the desired LSPs is accomplished by writing the appropriate values into the mode registers. The mode registers are accessed over the backplane TAP when the boundary scan controller issues the MODESEL, MODESEL1, or MODESEL2 instruction to the SCANSTA112. Note that at this point in the configuration sequence, when no LSPs have yet been unparked, the scan chain comprises only the SCANSTA112.

The effect of the values written to the various mode registers is discussed in Section 8.1. Writing the appropriate values into the mode registers selects the desired LSPs, but the selected LSPs are still parked at this point in the configuration sequence. In order to begin boundary scan operations the boundary scan controller issues the UNPARK instruction to the SCANSTA112 over the backplane TAP. As before, at this point in the configuration sequence the scan chain comprises only the SCANSTA112.

Once the backplane TAP sequences through the *Update-Instruction-Register* state after the UNPARK instruction is received, the selected LSPs are ready to be unparked. The LSPs actually enter the unparked state when the TAP controller sequences through the *Run-Test/Idle* state. At this point the LSP TMS lines and the state of the LSP TAP controllers begin to follow the backplane TAP. The TDO and TDI lines of the selected LSPs are connected through the SCANSTA112 to form the desired boundary scan chain. The configuration of the scan chain at this point depends upon which LSPs have been selected. This is shown in more detail in Table 8.

Upon completion of the required actions described by the Level 1 and 2 protocols the SCANSTA112 is in its operational state. When the SCANSTA112 is in its operational state, boundary scan data (sometimes referred to as scan vectors) are moved from the backplane TAP to one or more selected local scan chains.

In ScanBridge primary operating mode, the boundary scan controller must address the SCANSTA112, configure the local scan ports, and issue the UNPARK instruction over the backplane TAP prior to the start of boundary scan operations on the local scan chains. These initial instructions only affect the operation of the SCANSTA112 itself. These initial instructions can be inserted into the boundary scan test sequence automatically by most commercially available Automatic Test Pattern Generation (ATPG) tools. In an ATPG tool, this is called ScanBridge support.

The ScanBridge primary operating mode provides the same functionality as the (obsolete) SCANPSC110 and the SCANSTA111. These devices do not include a mode analogous to the stitcher mode.

When the SCANSTA112 is operated in stitcher mode it is always addressed, so the Level 1 and Level 2 protocols referred to above are not used. In stitcher mode the SCANSTA112 is always addressed and is always in its operational state. The relation of stitcher mode to the Level 1 and Level 2 protocols is described next.

3.5 Stitcher Mode

Stitcher mode provides a method of skipping the requirements of the Level 1 and 2 protocols of the ScanBridge mode of operation. If the SB/$\overline{\text{S}}$ pin is set to logic 0 (low) then the SCANSTA112 will be set to stitcher mode upon power up or reset.

After the SCANSTA112 is powered up or reset in stitcher mode it is not necessary to address it. In stitcher mode the SCANSTA112 always behaves as if it has been addressed. When in stitcher mode the SCANSTA112 goes directly to the operational state.

The stitcher mode is intended to permit the user to utilize the multiplexing capability of the SCANSTA112 with vector generation and delivery tools that do not feature ScanBridge support. A sequence generated by an ATPG tool without ScanBridge support may be used in a scan chain that includes a SCANSTA112 by using stitcher mode. When a SCANSTA112 is used in stitcher mode, such a boundary scan sequence may be used as is, without manual edits. Manual edits would be required to use the same sequence in a scan chain including a SCANSTA112 operating in ScanBridge mode.

In stitcher mode the selection of active LSPs is controlled by the LSP select register. This register is loaded at power up or reset with the logic values present on the LSPsel$_{(0–6)}$ pins. After the SCANSTA112 is powered up or reset in stitcher mode, the selected LSPs will be unparked and their TAP controllers will immediately follow the backplane TAP.

Mode registers 0-2 are not used to select LSPs when the SCANSTA112 is operated in stitcher mode. The desired LSPs are selected by the logic levels at the LSPsel$_{(0–6)}$ pins which set the values in the LSP select register. It is possible to change the contents of the LSP select register by writing directly to the register if it is accessible. This will change the LSP selection.

When the LSP selection is changed, local scan chains that are no longer selected will be parked in the *Run-Test/Idle* state after cycling through the *Test-Logic-Reset* state. Local scan chains that were previously parked will be re-synchronized with the backplane TAP and unparked when the backplane TAP sequences through the state in which they were parked.

The registers of the SCANSTA112 are accessible when the SCANSTA112 is set to normal, as opposed to full transparent, scan chain mode (scan chain modes are described in Section 4). In normal scan chain mode the registers of the SCANSTA112 can be written or read via the backplane TAP whether the SCANSTA112 is in stitcher or ScanBridge primary operating mode.

Writing the mode registers using the MODESEL, MODESEL1, and MODESEL2 instructions has no effect when the SCANSTA112 is in stitcher mode.

In stitcher mode, transparent scan chain mode is initiated by writing the TRANS bit of the control register rather than by using the TRANSPARENTENABLE and TRANSPARENTn instructions used in ScanBridge mode. (Scan chain modes are described in Section 4).

When the SCANSTA112 is used in stitcher mode it is normally configured by means of the device pins rather than by instructions from the boundary scan controller. This is the primary utility of stitcher mode. It should be noted, however, that many of the SCANSTA112 instructions available in ScanBridge mode are still available in stitcher mode. This gives the system designer increased flexibility in using the SCANSTA112.

In addition to reading and writing registers of the SCANSTA112, all the instructions that park and unpark the LSPs are available when the SCANSTA112 is in stitcher mode as long as it is not in full transparent mode.

There are differences in the operation of some instructions in stitcher mode. For example, while the SCANSTA112 is in stitcher mode the PARKTLR instruction will move the LSP TAPs to the *Test-Logic-Reset* state but will then idle them in the *Run-Test/Idle* state. The LSP TAPs will remain in the *Run-Test/Idle* state until the SCANSTA112 receives the UNPARK instruction and the backplane TAP is sequenced through the *Run-Test/Idle* state. The TLR_TRST and TLR_TRST_6 pins have no effect when the PARKTLR instruction is used in stitcher mode because the local scan chains are really parked in *Run-Test/Idle*. The operation of the TLR_TRST and TLR_TRST_6 bits in the control register and the associated SCANSTA112 pins is described in Section 7.5.

4 SCANSTA112 Scan Chain Modes

In each of the two primary operating modes, ScanBridge mode and stitcher mode, two scan chain modes are available. These are normal (not transparent) mode and full transparent mode. In ScanBridge primary operating mode an additional scan chain mode called transparent LSP mode is available.

The selected scan chain mode determines the configuration of the scan chain as seen from the boundary scan controller. We first describe the effect of each of these scan chain modes on the scan chain configuration and on the SCANSTA112.

4.1 Normal Scan Chain Mode

In normal scan chain mode the SCANSTA112 is itself part of the scan chain. This implies the following.

1. A register in the SCANSTA112 is inserted in the scan chain between the backplane TDI input and the first selected LSP TDO output. This register may be the instruction register, the boundary register, the bypass register, or any other register which can be inserted into the scan chain by the SCANSTA112.
2. The SCANSTA112 TAP controller sequences through the same series of states as the TAP controllers of the selected LSPs. When a five-high TMS reset is issued by the boundary scan controller, for example, the SCANSTA112 is also set to *Test-Logic-Reset* state. This will normally reset all the registers in the SCANSTA112 to their default values. Sometimes this reset behavior is undesired. A bit in the control register can be set to prevent unwanted resets of the SCANSTA112. The effects of this IgnoreReset bit will be described in more detail later.
3. A one-bit register is inserted between the TDI input of each LSP and the TDO output to the next LSP when multiple LSPs are chained together. This one-bit register re-synchronizes the LSP TDO transitions to the backplane TCK transitions on each pass through the SCANSTA112. It also delays the data sequence passing from the TDI of one LSP to the TDO of the next by one TCK period.
4. A single bit register is inserted between the TDI input of the final LSP in the chain and the TDO output to the backplane. This one-bit register re-synchronizes the backplane TDO transitions to the backplane TCK transitions. It also delays the data sequence passing from the TDI of the final LSP to the backplane TDO by one TCK period.

When the SCANSTA112 is in normal scan chain mode the scan chain may take on the following configuration:

Instruction Shift: TDI_B → SCANSTA112 IR (8-bit) → LSP_{B1} → PAD → LSP_{01} → PAD →LSP_n → PAD → TDO_B

Data Shift: TDI_B → SCANSTA112 DR → LSP_{B1} → PAD → LSP_{01} → PAD → LSP_n → PAD → TDO_B

Note that in these examples, with LSP_{B0} is selected as the master backplane TAP and LSP_{B1} operates as LSP_{00}.

In normal scan chain mode the SCANSTA112 can be controlled via the backplane TAP. If the boundary scan controller is to issue an instruction to the SCANSTA112 while in this mode, it sequences the TAP controller to the *Shift-Instruction-Register* state. In this state the instruction register of the SCANSTA112 and the instruction registers of all the devices on the selected LSPs are placed in the scan chain. The boundary scan controller can now shift in the instructions required to perform the desired boundary scan operation through the backplane TDI pin of the SCANSTA112. This can include instructions meant for the SCANSTA112 itself.

The SCANSTA112 is located at the beginning of the scan chain so instructions destined for the SCANSTA112 are shifted in at the end of the instruction sequence. When the boundary scan controller subsequently sequences the TAP controller to the *Update-Instruction-Register* state the instructions shifted into all the instruction registers, including the instruction register of the SCANSTA112, are captured and executed. In this mode the boundary scan controller can control the operation of the SCANSTA112 just as it controls the operation of all the other devices in the scan chain.

The insertion of the pad bit between the TDI input of the final LSP in the chain and the TDO output to the backplane insures that, from the perspective of the boundary scan controller, data transitions on the backplane TDO line occur on the falling edges of the backplane TCK. Figure 4 shows a timing diagram illustrating this point. In a basic boundary scan chain with no SCANSTA112, the TDO lines of all the devices in the scan chain change state on the falling edge of the TCK line. The TCK line is supplied to all the devices in parallel, so the transitions of each TDO line occur at approximately the same time.

When the SCANSTA112 is inserted in the scan chain it buffers the TCK line to all the selected LSP TCK lines. This introduces a delay between the backplane TCK signal and the TCK signal seen by all the devices in the LSP scan chain. These devices change their TDO lines on the falling edges of the TCK signal they see, which comes from the LSP of the SCANSTA112. At the TDO output of the SCANSTA112 there may be sufficient delay between the falling edge of the backplane TCK signal, which is being driven by the boundary scan controller, and the transitions of the TDO output to violate the timing conditions of the boundary scan controller.

Assuming that its own setup and hold time margins are met, the SCANSTA112 can re-time its TDO data to the backplane TCK. It detects the input data on the TDI line from the final LSP on the rising edge of the backplane TCK signal. This input data will have been set up by the last device in the local scan chain on the previous falling edge of the LSP TCK signal, which is delayed from the backplane TCK signal. The SCANSTA112 then transfers this data to the backplane TDO output on the **next** falling edge of the backplane TCK line. The result, from the viewpoint of the boundary scan controller, is that the backplane TDO line from the SCANSTA112 has its transitions aligned with the falling edges of the backplane TCK line. This provides setup and hold times, again from the point of view of the boundary scan controller, approximately equal to 1/2 clock period even though the setup time from the point of view of the SCANSTA112 may have been somewhat less than 1/2 clock period. The TDO data to the backplane is delayed by one bit from the TDO data from the last device in the scan chain. This is the pad bit added by the SCANSTA112 to re-synchronize the data.

Using the SCANSTA112 in normal mode requires that the boundary scan controller insert bits into the scan data stream to fill up the selected register in the SCANSTA112 and the pad bits inserted in the scan chain. This may require the user to modify an automatically generated test sequence with a text editor — or perhaps even with a hex editor. This may be something the user does not wish to undertake. Full transparent mode can be used to simplify the test sequence requirements. It may be necessary in full transparent mode to reduce the TCK frequency in order to satisfy the setup and hold time requirements of the boundary scan controller.

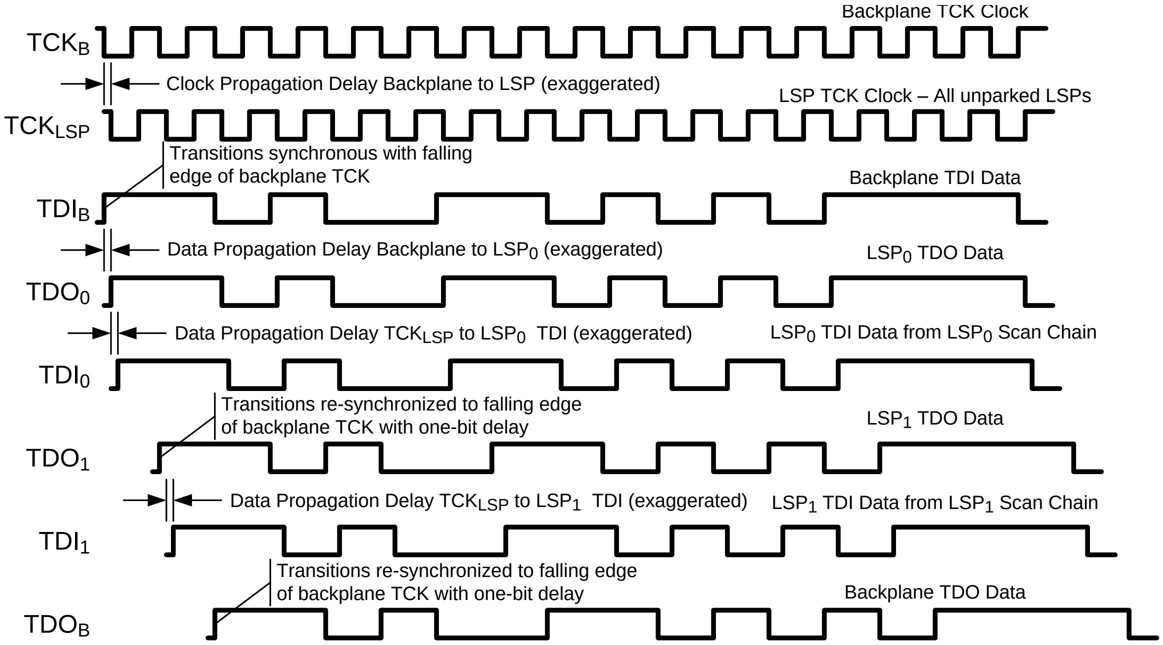
Backplane TCK Clock
TCK_B
Clock Propagation Delay Backplane to LSP (exaggerated)
LSP TCK Clock – All unparked LSPs
TCK_LSP
Transitions synchronous with falling edge of backplane TCK
Backplane TDI Data
TDI_B
Data Propagation Delay Backplane to LSP0 (exaggerated)
LSP0 TDO Data
TDO0
Data Propagation Delay TCK_LSP to LSP0 TDI (exaggerated)
LSP0 TDI Data from LSP0 Scan Chain
TDI0
Transitions re-synchronized to falling edge of backplane TCK with one-bit delay
LSP1 TDO Data
TDO1
Data Propagation Delay TCK_LSP to LSP1 TDI (exaggerated)
LSP1 TDI Data from LSP1 Scan Chain
TDI1
Transitions re-synchronized to falling edge of backplane TCK with one-bit delay
Backplane TDO Data
TDO_B

Figure 4. Timing Diagram – TDO_B/TCK_B Re-synchronization

4.2 Full Transparent Scan Chain Mode

In full transparent scan chain mode there are no pad bits or SCANSTA112 registers in the scan chain. Full transparent mode allows for the use of test sequences that were generated for a scan chain without the SCANSTA112 register and the pad bit. Such test sequences might come from an automatic test pattern generator or from software that generates programming sequences for a programmable logic device without ScanBridge support. When the SCANSTA112 is operated in full transparent scan chain mode such test sequences may be used without modification.

As an example, when the SCANSTA112 is operated in full transparent scan chain mode the scan chain may take on the following configuration:

TDI_B → LSP_{B1} → LSP_{01} → LSP_n → TDO_B

Note that in this example LSP_{B0} is selected as the backplane TAP and LSP_{B1} operates as LSP_{00}.

Conversely, as indicated above, when the SCANSTA112 is in normal scan chain mode the scan chain may take on the following configuration:

Instruction Shift: TDI_B → SCANSTA112 IR (8-bit) → LSP_{B1} → PAD → LSP_{01} → PAD →LSP_n → PAD → TDO_B

Data Shift: TDI_B → SCANSTA112 DR → LSP_{B1} → PAD → LSP_{01} → PAD → LSP_n → PAD → TDO_B

In full transparent scan chain mode the SCANSTA112 register does not appear in the scan chain. The pad bits also do not appear. The scan chain, from the perspective of the boundary scan controller, appears to consist only of the selected local scan chains. Note that any one, or more than one, local scan chains can be selected prior to initiating the full transparent scan chain mode of the SCANSTA112. The multiplexer function of the SCANSTA112 is thus still available.

When full transparent scan chain mode is activated, the selected LSP_n lines will follow the backplane TAP lines. $\overline{TRST}_{\overline{n}}$ will be a buffered version of $\overline{TRST}_{\overline{B}}$, TCK_n will be a buffered version of TCK_B, TMS_n will be a buffered version of TMS_B, TDO_n for the first LSP will be a buffered version of TDI_B and TDO_B will be a buffered version of TDI_n for the last LSP.

$\overline{\text{TRIST}}_{B0}$, $\overline{\text{TRIST}}_{B1}$, and $\overline{\text{TRIST}}_{(0-3)}$ will be asserted when the state machine is not in either the *Shift-Data-Register* or *Shift-Instruction-Register* states. This is consistent with the IEEE 1149.1 standard. The unselected LSPs will be placed in the *Parked-TLR* state, and their clocks will be gated (driven to a static 0) after 512 TCK_B clock cycles.

When the SCANSTA112 is in ScanBridge primary operating mode and one of the transparent mode instructions (TRANSPARENTENABLE or TRANSPARENTn) is shifted into the instruction register, the SCANSTA112 enters full transparent scan chain mode. After the instruction is scanned in, the tap controller goes through the *Update-Instruction-Register* state at which point $\overline{\text{TRST}}_{\overline{n}}$ will go high, and TMS_n will go low. This will force the targets connected to the LSP_n ports to go into the *Run-Test/Idle* state. Then, when the SCANSTA112 state machine goes into the *Run-Test/Idle* state, all of the LSP_n signals will follow the backplane TAP signals. This is identical to the method that is typically used to unpark a LSP. The TAP controller must be sequenced through the state in which the LSPs are parked in order to unpark them and allow them to follow the backplane TAP state.

The SCANSTA112 will remain in transparent mode until it is reset. Once in full transparent mode, the SCANSTA112 will not respond to a five-high TMS reset but it will respond to assertion of the $\overline{\text{TRST}}_{\overline{B}}$ line if the IgnoreReset bit in the control register is not set.

The SCANSTA112 can be used in a scan chain mode that provides a compromise between normal scan chain mode and full transparent scan chain mode. This is the transparent LSP scan chain mode, which is discussed next.

4.3 Transparent LSP Scan Chain Mode

Transparent LSP scan chain mode provides improved scan chain performance relative to both normal mode and full transparent mode. In this mode no pad bits are placed in the chain between unparked LSPs that are chained together. A register in the SCANSTA112 and the pad bit of the final selected LSP are included in the scan chain. This mode offers two advantages over full transparent mode: it is capable of higher speeds than full transparent mode because the final pad bit resynchronizes the data stream as it does in normal scan chain mode; and the boundary scan controller can control the SCANSTA112 via the backplane TAP in this mode.

In transparent LSP mode the chain may take on the following configuration:

Instruction: $\text{TDI}_B \rightarrow$ SCANSTA112 IR (8-bit) $\rightarrow \text{LSP}_a \rightarrow \text{LSP}_n \rightarrow$ PAD $\rightarrow \text{TDO}_B$

Data Shift: $\text{TDI}_B \rightarrow$ SCANSTA112 DR $\rightarrow \text{LSP}_a \rightarrow \text{LSP}_n \rightarrow$ PAD $\rightarrow \text{TDO}_B$

Note that in the transparent LSP mode no additional pad bits are inserted between LSPs that are chained together. Compare the chain configurations shown above to those for normal scan chain mode and full transparent scan chain mode.

The transparent LSP scan chain mode is only available when the SCANSTA112 is operating in its ScanBridge primary mode. In ScanBridge mode LSP selection is controlled by the values in mode registers 0-2. Transparent LSP scan chain mode is enabled by setting mode register 1 bit 7 to a 1. This bit has no effect when the SCANSTA112 is in stitcher mode.

Using the transparent LSP mode may require manual edits to boundary scan test sequences generated using ATPG tools. The SCANSTA112 register contents and the trailing pad bit can be inserted easily into a test sequence described by a Serial Vector Format (SVF) file. The additional bits can be set to static values. For example, the instruction to the SCANSTA112 can always be FF hex (11111111), which is the opcode for the BYPASS instruction. The required static bit patterns are specified by the HIR (Header-Instruction-Register), HDR (Header-Data-Register), TIR (Trailer-Instruction-Register), and TDR (Trailer-Data-Register) instructions in the SVF file. These bit patterns are automatically prepended or appended to each boundary scan transaction by the boundary scan controller so they do not need to be added explicitly to each boundary scan transaction. In normal mode when multiple LSPs are chained together, it may be necessary to edit each boundary scan transaction in order to put the pad bits in the right place in the bit sequence because there is a pad bit between each LSP's TDI and the TDO of the following LSP.

5 Primary Operating Modes and Scan Chain Modes

Both normal scan chain mode and transparent scan chain mode can be used in either the ScanBridge or stitcher primary operating mode. The interaction of the scan chain mode and the primary operating mode is described in this section.

5.1 Scanbridge Normal Scan Chain Mode

In order for the SCANSTA112 to operate in ScanBridge primary operating mode the SB/$\overline{\text{S}}$ bit in the control register must be set to 1. Since this bit is set to the value of the SB/$\overline{\text{S}}$ pin on power up or reset, this pin must be high when the power up or reset occurs in order for the SCANSTA112 to go into the ScanBridge mode by default. When the SCANSTA112 is powered up or reset in ScanBridge mode the values on the TRANS and LSPsel$_{(0–6)}$ pins are ignored. The SCANSTA112 must be configured via the backplane TAP in ScanBridge mode.

Once the SCANSTA112 has been addressed, selection of the LSPs is performed by writing the values of the mode registers 0-2. These registers are inserted into the boundary scan chain by writing the MODESEL, MODESEL1, or MODESEL2 instruction to the SCANSTA112. The selected LSPs become unparked when the backplane TAP controller cycles through the *Run-Test/Idle* state after the boundary scan controller has issued the UNPARK instruction to the SCANSTA112. In ScanBridge mode both the setting of the LSP select register and the LSPsel$_{(0–6)}$ pins are ignored.

5.2 Scanbridge Full Transparent Scan Chain Mode

In ScanBridge mode the SCANSTA112 must be addressed and its LSPs selected using the Level 1 and 2 protocols describe earlier. To use full transparent scan chain mode with the ScanBridge primary operating mode, a single SCANSTA112 must first be selected according to the Level 1 protocol and one or more LSPs must be selected according to the Level 2 protocol. This requires that the instruction sequences required by the Level 1 and Level 2 protocols be issued by the boundary scan controller over the backplane TAP.

Once a SCANSTA112 has been selected and its LSPs selected and unparked, an instruction to the SCANSTA112 via the backplane TAP enables transparent scan chain mode. The TRANSPARENTn or TRANSPARENTENABLE instruction will enable transparent scan chain mode for a SCANSTA112 in ScanBridge mode.

The TRANSPARENTENABLE instruction puts the SCANSTA112 in transparent scan chain mode with the scan chain defined by the contents of the mode registers. The TRANSPARENTn instruction puts the SCANSTA112 in transparent scan chain mode with a single LSP, LSP$_n$, overriding the contents of the mode registers.

Transparent scan chain mode will override any other active mode. In particular, it will override transparent LSP scan chain mode.

In transparent scan chain mode the TAP signals on the selected LSP(s) become buffered versions of the backplane signals. In this mode the TAPs on the LSPs can be reset by a five-high TMS reset and the SCANSTA112 will not respond to the reset.

The sequence of operations to activate transparent scan chain mode on a single LSP by issuing the TRANSPARENTn instruction are as follows. The example uses LSP$_0$. The example assumes that the SCANSTA112s in the system are operating in ScanBridge mode.

1. Scan the address of the desired SCANSTA112 into the instruction register (address a SCANSTA112).
2. Scan the TRANSPARENT0 instruction into the instruction register to enable full transparent scan chain mode on LSP$_0$. Full transparent mode will be enabled when the TAP controller enters the *Run-Test/Idle* state at the end of this shift operation. At this point $\overline{\text{TRST}}_{\overline{0}}$, TDO$_0$, TMS$_0$ and TCK$_0$ become buffered versions of $\overline{\text{TRST}}_{\overline{B}}$, TDI$_B$, TMS$_B$ and TCK$_B$ and TDO$_B$ becomes a buffered version of TDI$_0$.

Transparent mode will persist until the SCANSTA112 is reset. The GOTOWAIT and SOFTRESET instructions will not work in this mode. The SCANSTA112 will respond to the $\overline{\text{TRST}}_{\overline{B}}$ or $\overline{\text{RESET}}$ signals and will be reset when either of these signals is asserted. When the SCANSTA112 is reset it returns to the default primary operating mode. If this default primary operating mode is ScanBridge mode the SCANSTA112 will also return to normal scan chain mode.

It is possible to mask the $\overline{\text{TRST}_\text{B}}$ reset so that the SCANSTA112 will not respond to it when in ScanBridge transparent mode. When the IgnoreReset bit in the control register is set the SCANSTA112 will not respond to the $\overline{\text{TRST}_\text{B}}$ reset. When the IgnoreReset bit is set, the TAPs on the LSPs can be reset by the $\overline{\text{TRST}_\text{B}}$ reset and the SCANSTA112 will not be reset.

If the IgnoreReset bit in the control register is to be set, it must be set before the SCANSTA112 is put into transparent scan chain mode. When the SCANSTA112 is in transparent scan chain mode the control register is not accessible.

The boundary scan controller cannot control the SCANSTA112 via any of the backplane TAP lines when the SCANSTA112 is in full transparent mode and the IgnoreReset bit is set. The only way to reset the SCANSTA112 in this condition is to assert the $\overline{\text{RESET}}$ pin low or to cycle the power. This pin was not available in the SCANSTA111, so the SCANSTA111 could always be reset in transparent mode by a $\overline{\text{TRST}_\text{B}}$ reset. For applications that expect the SCANSTA112 to behave in the same manner as the SCANSTA111 it is suggested the $\overline{\text{TRST}_\text{B}}$ and the $\overline{\text{RESET}}$ pins be connected together. When this connection is made, asserting the $\overline{\text{TRST}_\text{B}}$ reset will always reset the SCANSTA112 even if the IgnoreReset bit in the control register is set.

5.3 Stitcher Normal Scan Chain Mode

When a single SCANSTA112 in a system is operated in stitcher mode, its internal registers may still be accessible to the boundary scan controller. When the SCANSTA112 is in stitcher primary mode and in normal scan chain mode, the boundary scan chain consists of the currently selected internal SCANSTA112 register, plus the LSPs that are selected via the LSP select register, plus the pad bits associated with the selected LSPs.

When the SCANSTA112 is in stitcher primary operating mode, the selection of normal scan chain mode or full transparent scan chain mode is determined by the value of the TRANS bit in the control register. This value is initialized by the value on the TRANS pin of the SCANSTA112 at power up or reset. If the TRANS pin is low at power up or reset, the TRANS bit in the control register will be initialized to 0 and the SCANSTA112 will go into stitcher normal scan chain mode.

As previously described, the LSP select register contents are initialized as the value of the respective $\text{LSPsel}_{(0\text{-}6)}$ pins upon power up or reset. These values may be changed by selecting the LSP select register using the LSPSEL instruction and loading new values into this register via the TDO pin. This changes the scan chain configuration to match the new set of LSPs in the LSP select register.

The control register may be modified in the same manner as the LSP select register by issuing the CONTROLSEL instruction. Some of the bits in the control register, such as the TRANS bit, assume default values on power up or reset according to the values on the SCANSTA112 input pins (see Table 4). Setting the TRANS bit in the control register to 1 when the SCANSTA112 is in stitcher normal mode will cause it to go to stitcher full transparent mode.

5.4 Stitcher Full Transparent Scan Chain Mode

When the SCANSTA112 is in stitcher primary operating mode the TRANS bit in the control register controls the activation of transparent mode. In stitcher mode the selection of normal or full transparent mode may be made by setting the logic value on the TRANS pin. The logic value on this pin is detected and used to set the value of the TRANS bit in the control register when the device is powered up or reset.

If the TRANS pin on the SCANSTA112 is a 1 at power up or reset the SCANSTA112 will be set to transparent scan chain mode if it is in stitcher primary operating mode. The TRANS bit in the control register is still set by the value on the TRANS pin at power up or reset even when the SCANSTA112 is in ScanBridge primary operating mode, but the register value is ignored when the SCANSTA112 is in ScanBridge primary operating mode.

The SCANSTA112 can also be set to stitcher full transparent mode from stitcher normal mode. Setting the SCANSTA112 to transparent mode is accomplished by setting the TRANS bit in the control register to 1. This is only effective in stitcher mode. The TRANS bit is ignored in ScanBridge mode. This should only be done after the desired LSPs are selected and any other register changes are completed because after the SCANSTA112 is set to full transparent scan chain mode the LSP select register will no longer be accessible.

Stitcher transparent mode is used when the SCANSTA112 is to function as an pin-controlled JTAG multiplexer. No SCANSTA112 registers or pad bits are placed into the scan chain. This mode of operation allows the use of most automatically generated test sequences without the need to manually edit them.

Setting the TRANS bit in the control register to 1 places the SCANSTA112 into the stitcher transparent mode. Prior to setting this bit, when the SCANSTA112 is not in full transparent mode, its TAP controller is controlled by the backplane TMS line just as the LSP TAP controllers are. When the control register in the SCANSTA112 is written with a new value which sets the TRANS bit to 1, and the TAP controllers move into the *Update-Data-Register* state, the internal instruction, data, and pad bit registers of the SCANSTA112 are removed from the chain, putting it into transparent mode. The TAP controllers of all unparked local scan chains will be in the *Update-Data-Register* state and will follow the backplane TAP controller. Data scanned in on the TDO line when the SCANSTA112 is in transparent mode will be presented directly to the TDI line of the first selected local scan chain. The SCANSTA112 will no longer respond to boundary scan instructions on the backplane TAP.

As in ScanBridge transparent mode, the TAPs on the LSPs can be reset by way of the five-high TMS reset and the SCANSTA112 will not respond to the reset. When the IgnoreReset bit in the control register is set, the TAPs on the LSPs can be reset by way of $\overline{\text{TRST}_\text{B}}$ reset and the SCANSTA112 will not respond to the reset. Note that the IgnoreReset bit in the control register must be set prior to activating the transparent scan chain mode since the control register is not accessible once this mode is activated. In stitcher transparent mode, as in ScanBridge transparent mode, when the IgnoreReset bit is set the only method of resetting the SCANSTA112 is to assert the $\overline{\text{RESET}}$ pin low or to cycle the power.

Once the SCANSTA112 is put in transparent mode, this mode can only be exited by a reset. Assertion of the $\overline{\text{TRST}_\text{B}}$ pin will reset the SCANSTA112 if the IgnoreReset bit in the control register is not set. If this bit is set then the SCANSTA112 may only be reset by asserting $\overline{\text{RESET}}$ or by a power cycle. Whenever the SCANSTA112 is in transparent mode, five-high TMS resets are ignored.

After the SCANSTA112 is reset the default values of the control register and LSP select register bits will be set from the external device pins as they were on power up. If the SCANSTA112 SB/$\overline{\text{S}}$ pin is still held low at this point, the SCANSTA112 will again enter stitcher mode. Depending upon the value at the TRANS pin, which is transferred to the TRANS bit in the control register, the SCANSTA112 will enter either normal or transparent stitcher mode. It is therefore possible for the SCANSTA112 to be reset in transparent stitcher mode and immediately enter transparent stitcher mode again, depending upon the values imposed at the device pins.

5.5 Primary Operating Mode/Scan Chain Mode Summary

The interactions between the device pins, the two primary operating modes, and the three scan chain modes is somewhat complex. The interactions of the primary operating and scan chain modes, the SCANSTA112 pins, and the SCANSTA112 registers are summarized in Table 1

Table 1. Primary Operating Mode Summary

Primary Operating Mode	ScanBridge Mode	Stitcher Mode
SB/$\overline{\text{S}}$ pin	Sets SB/$\overline{\text{S}}$ bit in control register on power up or reset	Sets SB/$\overline{\text{S}}$ bit in control register on power up or reset
SB/$\overline{\text{S}}$ bit in control register	1 sets ScanBridge mode	0 sets stitcher mode
$S_{(0–7)}$ slot address pins	Used for address matching (Level 1 protocol)	Not used
TRANS pin	Sets TRANS bit in control register on power up or reset	Sets TRANS bit in control register on power up or reset
TRANS bit in control register	Ignored	1 sets full transparent mode 0 sets normal mode
$\text{LSPsel}_{(0–6)}$ pins	Sets LSP select register on power up or reset	Sets LSP select register on power up or reset
LSP select register	Ignored	Sets selected LSPs
Mode registers 0-2	Sets selected LSPs, transparent LSP mode, and GPIO settings	Ignored
Transparent LSP mode	Mode available	Mode not available
Registers in normal scan chain mode	Accessible	Accessible

Table 1. Primary Operating Mode Summary (continued)

Primary Operating Mode	ScanBridge Mode	Stitcher Mode
Registers in full transparent scan chain mode	Not accessible	Not accessible
Registers in transparent LSP scan chain mode	Accessible	Mode not available
Pad bits in normal scan mode	Inserted at the end of each local scan chain	Inserted at the end of each local scan chain
Pad bits in full transparent scan chain mode	None	None
Pad bits in transparent LSP scan chain mode	Inserted after the final local scan chain only	Mode not available

6 Operational Mode and Reset Summary

In a boundary scan chain a reset is not the extraordinary event it might be in an operational system. The TAP controllers of most of the devices in any boundary scan chain spend most of their time in the *Test-Logic-Reset* state. When the TAP controller is in the *Test-Logic-Reset* state the system is ready to perform its normal operations.

In a system with one or more SCANSTA112s it will frequently be necessary to reset them. The SCANSTA112s will normally be reset when the boundary scan operations are complete. They will also be reset when the operational mode changes, when the LSPs are to be re-synchronized, or when a different SCANSTA112 or group of SCANSTA112s is to be selected. Accordingly it is important to describe the types of reset available in a scan chain including the SCANSTA112 and how they interact with the various operational modes of the SCANSTA112.

The first reset method is power up or $\overline{\text{RESET}}$. These are actually two reset methods, of course, but they both have the same effect on the SCANSTA112. When power is first applied or the $\overline{\text{RESET}}$ pin is asserted (driven to a logic 0), the SCANSTA112 will execute a complete reset.

The second reset method is assertion of the $\overline{\text{TRST}}_{\text{B}}$ pin. This is the optional asynchronous reset specified in the IEEE 1149.1 standard. Assertion of the $\overline{\text{TRST}}_{\text{B}}$ pin will also normally cause the SCANSTA112 to execute a complete reset. The action of this pin may be masked as shown in Table 2 and Table 3.

The final reset method is what is called a five-high TMS reset. This reset method exploits the fact that the IEEE 1149.1 TAP controller state machine will go to the *Test-Logic-Reset* state when the TMS line is held high for five TCK cycles regardless of the state in which it starts. A five-high TMS reset will also normally cause the SCANSTA112 to execute a complete reset. This reset mechanism may be gated by the scan chain mode and by the IgnoreReset pin as shown in Table 2 and Table 3.

When any complete reset occurs all the SCANSTA112 registers are reset to their default initial values. Registers whose initial values are controlled by external pins will be set according to the values on the corresponding pins as in Table 4. Register bits not defined in Table 4 will be set to an initial value of zero when the SCANSTA112 is powered up or reset.

Table 2. Reset effect when IgnoreReset is deasserted (set to 0)

IgnoreReset = 0	ScanBridge Mode	Transparent Scan Chain Mode [(1)]	Stitcher Mode
$\overline{\text{RESET}}$	Full Reset	Full Reset	Full Reset
$\overline{\text{TRST}}_{\text{B}}$	Full Reset	Full Reset	Full Reset
5 high TMS	Full Reset	Ignored	Full Reset

[(1)] Transparent scan chain mode is available in both ScanBridge and stitcher primary operating modes

Table 3. Reset effect when IgnoreReset is asserted (set to 1)

IgnoreReset = 1	ScanBridge Mode	Transparent Scan Chain Mode [1]	Stitcher Mode
$\overline{\text{RESET}}$	Full Reset	Full Reset	Full Reset
$\overline{\text{TRST}}_B$	Ignored	Ignored	Ignored
5 high TMS	Ignored	Ignored	Ignored

[1] Transparent scan chain mode is available in both ScanBridge and stitcher primary operating modes

Table 4. Initial Register Values

Register	External Pins	Internal Value
Instruction Register	N/A	XXH ID code
Mode Register 0	N/A	01H Select LSP$_{00}$
Control Register	MPsel$_{B1/\overline{B0}}$, SB/$\overline{\text{S}}$, TRANS, TLR_TRST, TLR_TRST_6	0b IgnoreReset
LSP Select Register	LSPsel$_{(0-6)}$	

It is often desired in boundary scan operations to reset the scan chain on one or more LSPs of the SCANSTA112 without resetting the SCANSTA112 itself. This might be necessary if, for example, a single SCANSTA112 has already been addressed and its desired LSPs selected. The SCANSTA112 will normally be fully reset by assertion of the $\overline{\text{TRST}}_B$ pin just as it would be by assertion of the $\overline{\text{RESET}}$ pin. In order to enable the SCANSTA112 to retain its configuration while asynchronously resetting the devices on its LSPs, the $\overline{\text{TRST}}_B$ pin may be masked by the control register IgnoreReset bit. If this bit is set the SCANSTA112 TAP controller will cycle to the *Test-Logic-Reset* state but the SCANSTA112 configuration registers will not be reset. This permits the user to propagate a $\overline{\text{TRST}}_B$ reset pulse through the SCANSTA112 without resetting the Level 1 and 2 protocols.

Asserting the $\overline{\text{TRST}}_B$ pin with the IgnoreReset bit of the control register set will assert the $\overline{\text{TRST}}_n$ pin on all unparked LSPs. This will place all unparked LSP TAP controllers in the *Test-Logic-Reset* state (if the devices are equipped with the optional $\overline{\text{TRST}}$ pin; see below). All parked LSP TAP controllers will be placed in the *Parked-TLR* state. If the SCANSTA112 is in normal scan chain mode or transparent LSP scan chain mode, its TAP controller will also be placed in the *Test-Logic-Reset* state but, as noted, the internal registers of the SCANSTA112 will not be reinitialized. If the SCANSTA112 is in transparent mode asserting the $\overline{\text{TRST}}_B$ pin will have no effect on the SCANSTA112 when the IgnoreReset bit in the control register is set.

When the IgnoreReset bit is set and a $\overline{\text{TRST}}_B$ is performed, target devices with $\overline{\text{TRST}}$ pins on unparked LSPs will receive a $\overline{\text{TRST}}$ assertion while target devices without a $\overline{\text{TRST}}$ pin will not. The SCANSTA112 does not automatically sequence the selected LSPs through a five-high TMS reset. If devices without a $\overline{\text{TRST}}$ pin are used in the system, the boundary scan controller should sequence the target TAPS back to the *Test-Logic-Reset* state using the five-high TMS reset to ensure synchronization of all the targeted TAP controllers.

The effect of the five-high TMS reset mechanism is controlled by the selection of scan chain mode and by the IgnoreReset bit in the control register. When the SCANSTA112 is not in full transparent mode, a five-high TMS reset will reset all TAP controllers including the TAP controller of the SCANSTA112 itself unless the IgnoreReset bit is set. If the IgnoreReset bit is set a five-high TMS reset will be ignored. In this case, as with the assertion of the$\overline{\text{TRST}}_B$ pin, the TAP controller of the SCANSTA112 will be placed in the *Test-Logic-Reset* state, but the internal registers of the SCANSTA112 will not be reinitialized.

When the SCANSTA112 is in full transparent scan chain mode, the five-high TMS reset is ignored. When the SCANSTA112 is in full transparent mode it is not part of the scan chain, so it does not respond to any events on the scan chain, including a five-high TMS reset.

In summary, when utilizing the SCANSTA112 there are times in which the user may wish to reset devices that are connected to the LSPs without resetting the SCANSTA112. The IgnoreReset bit in the control register provides a method of accomplishing this. The $\overline{\text{RESET}}$ pin is the ultimate reset and will override all register control. Other resets can be masked via the IgnoreReset bit in the control register.

Table 2 indicates the effect of the resets during the various modes of operation when the IgnoreReset bit is deasserted (set to 0). Table 3 indicates the effect of the resets during the various modes of operation when the IgnoreReset bit is asserted (set to 1).

We have now described in broad outlines all the necessary steps for using the SCANSTA112 as a JTAG multiplexer. We have described the procedures for selecting one or more SCANSTA112s; setting the primary operating mode for the SCANSTA112s; setting the scan chain mode for the SCANSTA112s; configuring the LSPs for the SCANSTA112s; and resetting the SCANSTA112s. Everything we have presented so far has been targeted at the user of the device and written to describe its behavior “from the pins out”.

A high-level description of the internal architecture of the SCANSTA112 can provide valuable additional insight for the system designer. The next section of this document presents a description of the SCANSTA112 “from the pins in”.

7 SCANSTA112 State Machines

The SCANSTA112 is compliant to the IEEE 1149.1 standard in that it supports all required IEEE 1149.1 operations. The SCANSTA112 extends the IEEE 1149.1 standard to a multi-drop boundary scan system. In this context a multi-drop system is one with more than one boundary scan chain. Multiple scan chains may be implemented by including multiple SCANSTA112s in a system, by using multiple LSPs on one or more SCANSTA112s, or by a combination of the two. Where multiple SCANSTA112s are used in a system, a higher level protocol, referred to as the Level 1 protocol, is used to address the desired set of SCANSTA112s.

In multi-drop scan systems a boundary scan controller can select individual SCANSTA112s for participation in upcoming scan operations. SCANSTA112 selection is accomplished by scanning a device address out simultaneously to multiple SCANSTA112s. Only the SCANSTA112 whose statically-assigned address matches the scanned-out address becomes selected to receive further instructions from the boundary scan controller. Alternatively a reserved address designed to select multiple SCANSTA112s simultaneously may be scanned out. The SCANSTA112s selected by this broadcast or multi-cast address become selected for boundary scan operations.

SCANSTA112 selection is accomplished according to the Level 1 protocol. Subsequent configuration instructions are sent to selected SCANSTA112s using the Level 2 protocol. The Level 2 protocol specifies instructions that configure the LSPs and other operational characteristics of the SCANSTA112 by means of instructions delivered to the SCANSTA112 via its TAP.

The SCANSTA112 control architecture comprises three distinct but coupled types of state machines (see Figure 5). The first is the SCANSTA112 selection state machine (see Figure 6) which implements the Level 1 protocol. The second state machine actually consists of seven identical but independent state machines (see Figure 10), one for each SCANSTA112 local scan port. Each of these local scan port configuration state machines controls the selection of one of the SCANSTA112's local scan ports. These state machines implement the Level 2 protocol. The third is the TAP controller state machine, which is used to control the operations of the SCANSTA112 boundary scan TAP according to the requirements of the IEEE 1149.1 Standard (see Figure 2). This state machine implements the operational mode of the SCANSTA112.

Note that when the SCANSTA112 is used in stitcher mode the first two state machines are bypassed and the LSPs selected by the value in the LSP select register are immediately placed in the *Unparked* state. Since the SCANSTA112 selection state machine is not active when the SCANSTA112 is in stitcher mode, we will assume as we describe the operation of this state machine that the SCANSTA112 is in ScanBridge mode.

7.1 SCANSTA112 Selection State Machine

The SCANSTA112 selection state machine performs the address matching which gives the SCANSTA112 its multi-drop capability. The SCANSTA112 selection state machine supports the addressing of a single SCANSTA112 from multiple SCANSTA112s in a system, multi-cast addressing of a group of SCANSTA112s simultaneously, and broadcast addressing of all the SCANSTA112s in a system simultaneously. Selecting a single SCANSTA112 is referred to as direct addressing. Selecting a SCANSTA112 as part of a group using either the broadcast address or a multi-cast address is called multiple addressing.

The SCANSTA112 selection state machine implements the SCANSTA112's Level 1 protocol. As mentioned, we will assume in what follows that the SCANSTA112 is operating in ScanBridge primary operating mode.

The IEEE 1149.1 boundary scan controller in a system sends instructions to a SCANSTA112 via the backplane TAP. On power up, or following a reset, the SCANSTA112's selection state machine is in the *Wait-For-Address* state. When the SCANSTA112's TAP controller is sequenced to the *Shift-Instruction-Register* state, data shifted in through the TDI_B input is shifted into the SCANSTA112's instruction register. Note that prior to successful selection of a SCANSTA112, data is not shifted out of the instruction register and out through the SCANSTA112's TDO_B output, as it is during normal scan operations. Instead, as each new bit enters the instruction register's most-significant bit position, data shifted out from the least-significant bit position is discarded. The TDO_B output is held in tri-state to prevent contention with other data.

To address a SCANSTA112, the boundary scan controller shifts an address into the SCANSTA112's instruction register. After the instruction register is loaded with the address data, the SCANSTA112's TAP controller enters the*Update-Instruction-Register* state. At this point, the SCANSTA112's address-recognition logic compares the address in the 8-bit instruction register with the 8-bit assigned slot address which is present on the $S_{(0-7)}$ inputs. Simultaneously, the scanned-in address is compared with the reserved broadcast and multi-cast addresses. If an address match is detected, the SCANSTA112 selection state machine enters one of the two selected states, either: *Selected-Single-SCANSTA112* if the address matches the assigned slot address; or *Selected-Multiple-SCANSTA112* if the address matches the broadcast address or the multi-cast address of the SCANSTA112's assigned multi-cast group (more on that later).

If the scanned address does not match the assigned slot address or one of the reserved broadcast or multi-cast addresses, the SCANSTA112 selection state machine enters the *Unselected* state. The SCANSTA112's selection state machine will remain in the *Unselected* state until the SCANSTA112 is reset or until the GOTOWAIT instruction is received. All the SCANSTA112s in a system, selected or not, will respond to the GOTOWAIT instruction by placing their selection state machines into the *Wait-For-Address* state unless they are in full transparent scan chain mode.

Once a SCANSTA112 has been selected, the Level 2 protocol is used to issue commands and to access the SCANSTA112's various registers. At this point the local scan port configuration state machines become active.

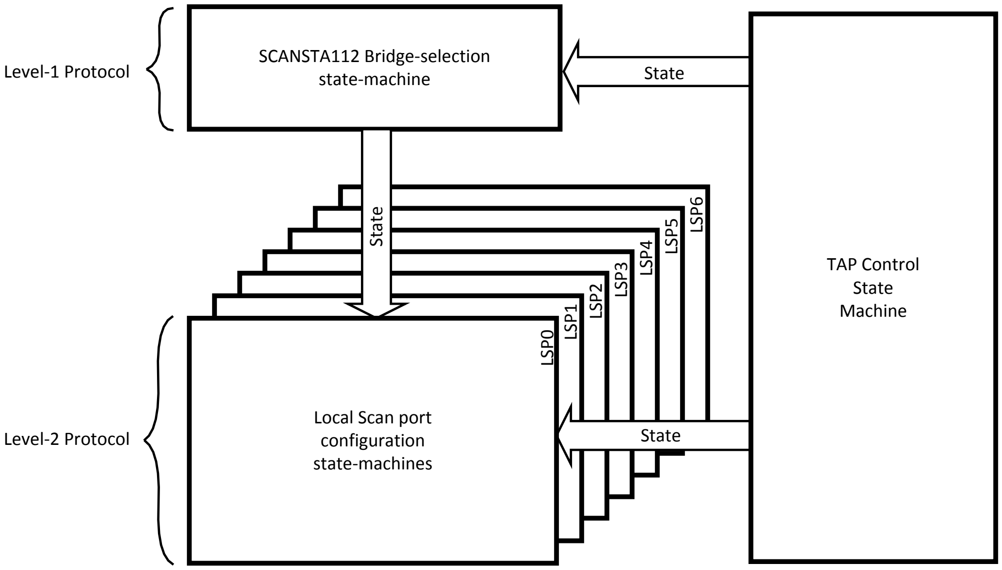
Level-1 Protocol
SCANSTA112 Bridge-selection state-machine
State
State
LSP6
LSP5
LSP4
LSP3
LSP2
LSP1
LSP0
TAP Control State Machine
Level-2 Protocol
Local Scan port configuration state-machines
State

Figure 5. SCANSTA112 State Machines

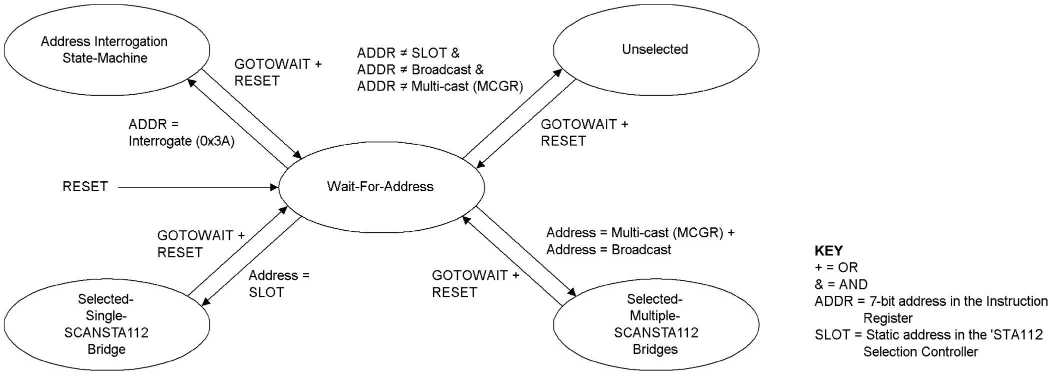
Address Interrogation State-Machine
GOTOWAIT + RESET
ADDR ≠ SLOT &
ADDR ≠ Broadcast &
ADDR ≠ Multi-cast (MCGR)
Unselected
ADDR = Interrogate (0x3A)
GOTOWAIT + RESET
RESET
Wait-For-Address
GOTOWAIT + RESET
Address = Multi-cast (MCGR) +
Address = Broadcast
Address = SLOT
GOTOWAIT + RESET
Selected-Single-SCANSTA112 Bridge
Selected-Multiple-SCANSTA112 Bridges
KEY
+ = OR
& = AND
ADDR = 7-bit address in the Instruction Register
SLOT = Static address in the 'STA112 Selection Controller

Figure 6. State Machine for SCANSTA112 Selection Controller

7.2 Multiple Addressing

Groups of SCANSTA112s, or all the SCANSTA112s in a system, can be simultaneously addressed. When addressed this way, all the selected SCANSTA112s define parallel active scan chains. This requires a departure from the behavior described in the IEEE 1149.1 standard to prevent contention between the TDO_B drivers of the SCANSTA112s which are all connected together to the backplane TDI line. This behavior is described below.

The reserved broadcast address can be issued by the boundary scan controller to simultaneously select all SCANSTA112s in a system. This broadcast mode is useful in systems which contain multiple identical boards. By addressing all the SCANSTA112s in a system by means of the broadcast address the boundary scan controller can perform boundary scan operations on all the identical boards in the system at once.

To avoid contention on the backplane TDI line between the TDO_B output drivers of different SCANSTA112s, each SCANSTA112's TDO_B driver is always at TRI-STATE while in broadcast mode. In this configuration, the on-chip Linear Feedback Shift Register (LFSR) can be used to accumulate a test result signature for each board. These signatures can be read back later by direct addressing each board's SCANSTA112. Note that this requires that the system have the ability to directly address each SCANSTA112, so unique static addresses for each SCANSTA112 must still be provided.

As a way to make the broadcast mechanism more selective, the SCANSTA112 provides a multi-cast addressing mode. A SCANSTA112's multi-cast group register (MCGR) can be programmed to assign that SCANSTA112 to one of four multi-cast groups. When SCANSTA112s in the *Wait-For-Address* state are updated with one of the four reserved multi-cast addresses, all SCANSTA112s whose MCGR matches the multi-cast group to which that address is assigned will become selected. As in broadcast mode, TDO_B is always at TRI-STATE while in multi-cast mode. Unique static addresses for each SCANSTA112 must be provided in order to perform the initial assignment of each SCANSTA112 to one of the multi-cast groups and also, as in broadcast mode, to read out the signature results from the LFSR after the boundary scan operations are performed.

Note that the slot address inputs should not be set to a value corresponding to a multi-cast group, to the broadcast address, or to the interrogation address. If a SCANSTA112's slot address is set to one of the reserved addresses it will not be directly addressed when that address is issued by the boundary scan controller.

As will be described in a later section, use of the address interrogation feature of the SCANSTA112 places some further restrictions on the available address space. If address interrogation is to be used in the system no SCANSTA112 should be assigned a slot address of 00 Hex (00000000). This address assignment conflicts with the address interrogation operation. Also, since the SCANSTA112 reports only the least-significant six bits of its slot address during address interrogation, only addresses in the range from 01 Hex to 39 Hex should be assigned if address interrogation is to be used.

Table 5. SCANSTA112 Address Modes

Address Type	Hex Address	Binary Address	TDO_B State
Direct Address	00 to 39,	00000000 to 00111001	Normal IEEE Std. 1149.1
	40 to FF.	01000000 to 11111111	
Interrogation Address	3A	00111010	Force strong "0" or weak "1" as ones-complement address is shifted out.
Broadcast Address	3B	00111011	Always at TRI-STATE
Multi-Cast Group 0	3C	00111100	Always at TRI-STATE
Multi-Cast Group 1	3D	00111101	Always at TRI-STATE
Multi-Cast Group 2	3E	00111110	Always at TRI-STATE
Multi-Cast Group 3	3F	00111111	Always at TRI-STATE

7.3 Addressing Modes

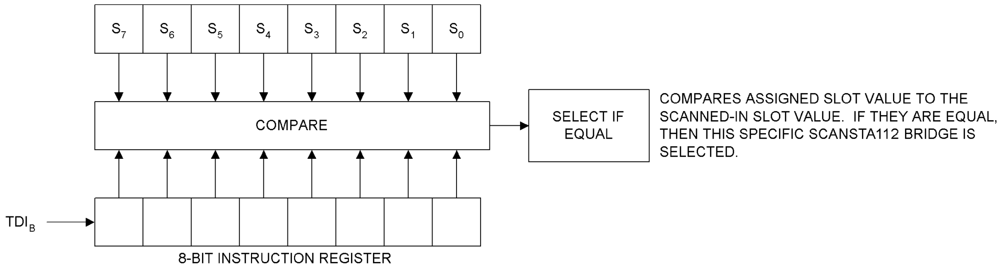
S_7
S_6
S_5
S_4
S_3
S_2
S_1
S_0
COMPARE
SELECT IF EQUAL
COMPARES ASSIGNED SLOT VALUE TO THE SCANNED-IN SLOT VALUE. IF THEY ARE EQUAL, THEN THIS SPECIFIC SCANSTA112 BRIDGE IS SELECTED.
TDI_B
8-BIT INSTRUCTION REGISTER

Figure 7. Direct Addressing: Device Address Loaded into Instruction Register

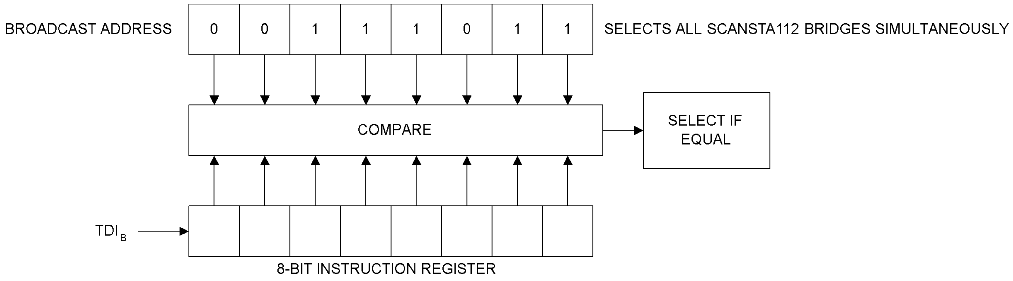
BROADCAST ADDRESS
0
0
1
1
1
0
1
1
SELECTS ALL SCANSTA112 BRIDGES SIMULTANEOUSLY
COMPARE
SELECT IF EQUAL
TDI_B
8-BIT INSTRUCTION REGISTER

Figure 8. Broadcast Addressing: Address Loaded into Instruction Register

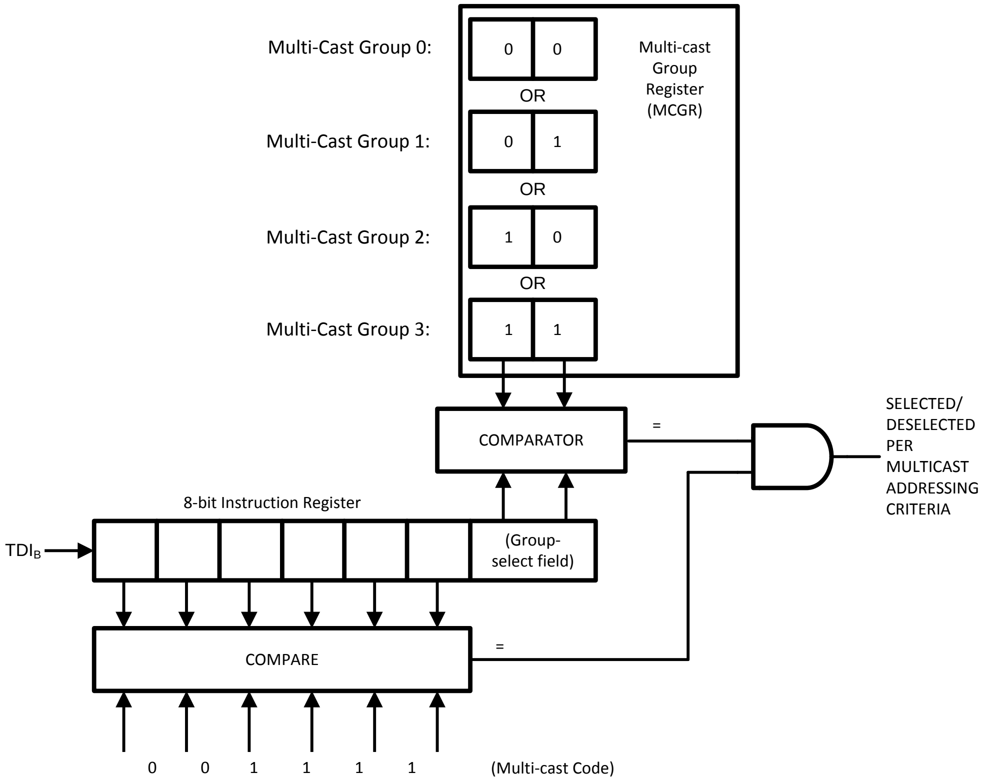

Figure 9. Multi-Cast Addressing: Address Loaded into Instruction Register

7.4 SCANSTA112 Local Scan Port Configuration State Machine

The SCANSTA112's local scan port configuration state machine consists of seven independent state machines. These state machines are used to control the insertion of local scan ports into the overall scan chain or the isolation of local ports from the chain. To simplify the nomenclature in the explanation that follows we will use the term LSP to refer to either a local scan port configuration state machine, its associated local scan port, or the local scan chain attached to that local scan port. Which of these is meant will be clear from the context.

From the perspective of the boundary scan controller in a system, a selected SCANSTA112 presents only one scan chain, called the active scan chain, regardless of how many local scan chains are connected to its LSPs. The SCANSTA112 architecture allows one or more of the SCANSTA112's local scan chains to be included in the active scan chain. The selected SCANSTA112 is also included in the active scan chain.

The SCANSTA112 receives instructions and data sent to it over the backplane TAP. The data and instructions are scanned out to the local scan chains on unparked LSPs that form the active scan chain. The scan chains on parked LSPs are not part of the active scan chain. They are held in a stable state and do not receive data over the LSP TAP.

Each LSP can be parked in one of four stable states: *Parked-TLR*, *Parked-RTI*, *Parked-Pause-DR* or *Parked-Pause-IR*. Parking an unparked scan chain removes that local scan chain from the active scan chain. The parked scan chain is held in its parked state until it is unparked by instructions issued to the SCANSTA112 to which it is attached. When a parked scan chain is unparked it is inserted into the active scan chain.

As shown in Figure 10, the SCANSTA112's seven local scan port configuration state machines allow each of the SCANSTA112's LSPs to occupy a different state at any given time. Some LSPs may be parked, perhaps in different stable states, while other LSPs are part of the active scan chain. The state diagram shows that some state transitions of the local scan port configuration state machine depend on the current state of the TAP control state machine. As an example, a LSP which is parked in the *Parked-RTI* state does not become unparked (*i.e.*, enter the *Unparked* state) until the SCANSTA112 receives an UNPARK instruction and the SCANSTA112's TAP controller state machine enters the *Run-Test/Idle* state.

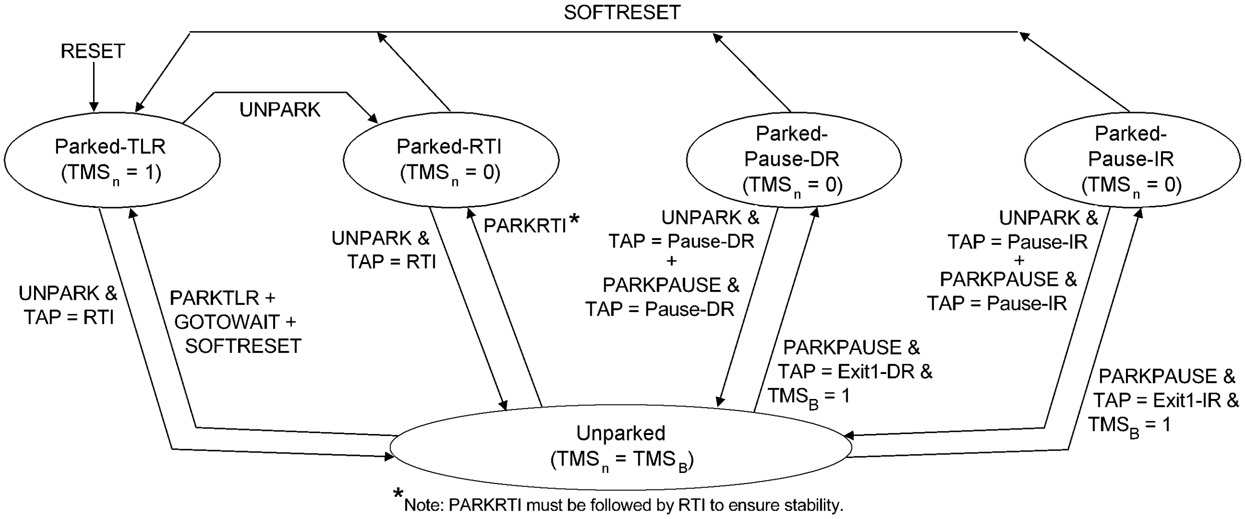
SOFTRESET
RESET
UNPARK
Parked-TLR (TMS$_n$ = 1)
Parked-RTI (TMS$_n$ = 0)
Parked-Pause-DR (TMS$_n$ = 0)
Parked-Pause-IR (TMS$_n$ = 0)
UNPARK & TAP = RTI
PARKTLR + GOTOWAIT + SOFTRESET
UNPARK & TAP = RTI
PARKRTI*
UNPARK & TAP = Pause-DR + PARKPAUSE & TAP = Pause-DR
PARKPAUSE & TAP = Exit1-DR & TMS$_B$ = 1
UNPARK & TAP = Pause-IR + PARKPAUSE & TAP = Pause-IR
PARKPAUSE & TAP = Exit1-IR & TMS$_B$ = 1
Unparked (TMS$_n$ = TMS$_B$)
*Note: PARKRTI must be followed by RTI to ensure stability.

Figure 10. Local SCANSTA112 Port Configuration State Machine

When a local scan port configuration state machine is in the *Unparked* state its LSP is part of the active scan chain. The TAP controller of its LSP follows the state of the backplane TAP controller. When the local scan port configuration state machine leaves the *Unparked* state its LSP is removed from the active scan chain. For example, when the SCANSTA112 receives the PARKRTI instruction it prepares to send the local scan port configuration state machines in the *Unparked* state to the *Parked-RTI* state. After this instruction is received, when the backplane TAP controller is transitioned through the *Run-Test/Idle* state, each unparked LSP enters the *Parked-RTI* state. In this state the TMS$_n$ line of the LSP will be held low, holding the local scan chain TAP controller in the *Run-Test/Idle* state until the LSP is later unparked.

These examples highlight the issue of TAP synchronization. A boundary scan chain always has one TAP controller state. The individual TAP controllers of all the devices in the scan chain are always in the same state. When more than one local scan chain exists in a system, and when local scan chains are inserted and removed from the active scan chain, it is necessary to insure that the TAP controllers of all the devices in the active scan chain are always in the same state, at least during data shift operations. This has implications for the interaction of the local scan port configuration state machine and the backplane TAP controller state machine.

The three park instructions, PARKTLR, PARKRTI, and PARKPAUSE, which may be issued to the SCANSTA112 are scanned into the backplane TAP when the backplane TAP controller is in the *Scan-Instruction-Register* state. The instructions become active once the TAP controller state machine enters the *Update-Instruction-Register* state. At this point the TAP controller is in one state (*Update-Instruction-Register*) and the LSPs are to be parked in another state. If the instruction received is PARKTLR then a modified five-high TMS reset sequence is applied to each unparked LSP. Since the TAP controller of the active scan chain, which includes the unparked LSPs, starts out in the *Update-Instruction-Register* state, only three clocks with TMS high are required to put the LSP in *Test-Logic-Reset* state. It is not possible for the SCANSTA112 to receive a new instruction that might unpark the parked LSPs within three clocks, so this is always safe. The backplane TAP may not be synchronized to the LSP TAPs for three clock pulses, but they can always be re-synchronized when necessary.

If the instruction received is PARKPAUSE then the behavior is different. This instruction is used to park the LSP TAPs in either the *Pause-Instruction-Register* or *Pause-Data-Register* state. If the backplane TAP controller transitions to the *Run-Test/Idle* state without passing through either of the pause states and stays there, the LSP TAP controller also remains in the *Run-Test/Idle* state. No operations occur on the TAP in the *Run-Test/Idle* state. The PARKPAUSE instruction has not been executed when this happens, but it remains the active instruction, ready to be executed when the TAP cycles through the appropriate state.

If the backplane TAP controller transitions to the *Test-Logic-Reset* state, the LSP's TAP controller will be reset. Assuming the SCANSTA112 is not reset because the IgnoreReset bit is set, the SCANSTA112's local scan port configuration state machine will still be ready to park the LSP in one of the pause states. Eventually, if boundary scan operations resume, the active scan chain TAP controller will be sequenced through either the *Exit1-Data-Register* or *Exit1-Instruction-Register* state. When either of these TAP states is detected, the LSP will be parked in the associated pause state and will remain there until it is unparked.

In summary, when the PARKPAUSE instruction is scanned into the SCANSTA112 the unparked LSPs will be parked in one of the stable pause states when the active scan chain TAP controller is sequenced through the *Exit1-Data-Register* or *Exit1-Instruction-Register* state. Until this happens the LSPs will remain part of the active scan chain.

When the PARKRTI instruction is updated into the instruction register the LSP TMS line is driven low. This puts the selected LSP TAP controller into the *Run-Test/Idle* state on the next TCK rising edge and holds it there. After the PARKRTI instruction is updated into the instruction register the backplane TAP controller must also be cycled to the *Run-Test/Idle* state in order to insure stability.

When the user desires to unpark a LSP that has previously been parked in any of the stable states it is necessary to sequence the backplane TAP through that stable state in order to re-synchronize the LSP TAP with the backplane TAP. A parked LSP will remain in its parked stable state until the UNPARK instruction is scanned into the instruction register of the SCANSTA112 and the backplane TAP controller is sequenced through the stable state in which the LSP is parked.

The SCANSTA112's local scan port configuration state machine implements the SCANSTA112's Level 2 protocol. The associated Level 2 protocol instructions to the SCANSTA112 are as follows.

- SOFTRESET – this instruction parks all LSPs, whether they are currently parked or unparked, in the *Test-Logic-Reset* state if the SCANSTA112 is in ScanBridge primary operating mode or in the *Run-Test/Idle* state if it is in stitcher operating mode. The local port configuration state machine is placed in the *Parked-TLR* state.
- PARKTLR – this instruction has the same effect on unparked LSPs as the SOFTRESET instruction. This instruction has no effect on parked LSPs.
- PARKRTI – this instruction parks all unparked LSPs in the *Run-Test/Idle* state. The local scan port configuration state machine is placed in the *Parked-RTI* state.
- PARKPAUSE – this instruction parks all unparked LSPs in the *Pause-Instruction-Register* or *Pause-Data-Register* state. The state in which the LSP is parked depends upon the state transitions of the backplane TAP controller. This command can also unpark LSPs parked in one of the pause states. Figure 10 illustrates the operation of the PARKPAUSE instruction and its interaction with the backplane TAP controller state.
- UNPARK – this instruction unparks all parked LSPs. Each LSP becomes unparked when the backplane TAP controller is sequenced through the state in which it was parked. As each LSP is unparked its local scan port configuration state machine enters the *Unparked* state.

These instructions are considered Level 2 protocol instructions because they directly affect the state of the local scan port configuration state machines. The SCANSTA112 instruction set includes other instructions which are described in Section 9. These instructions access registers in the SCANSTA112 and set its primary and scan chain operating modes. These instructions may affect the state of the local scan port configuration state machines as well as the Level 2 protocol instructions listed above. The instructions listed above directly set the states of the local scan port configuration state machines and the associated LSPs with no other effects. This is the reason for considering these instructions, and not others, to be Level 2 protocol instructions.

7.5 State Machine Interaction

Figure 11 illustrates how the SCANSTA112's state machines interact. The SCANSTA112 selection state machine enables or disables operation of the SCANSTA112's seven local scan port selection state machines. In SCANSTA112s which are selected via the Level 1 protocol, either by direct addressing or as members of broadcast or multi-cast groups, Level 2 protocol commands can be used to park or unpark local scan ports. Note that most transitions of the local scan port configuration state machines are gated by particular states of the SCANSTA112's TAP controller state machine, as shown in Figure 10 and Figure 11.

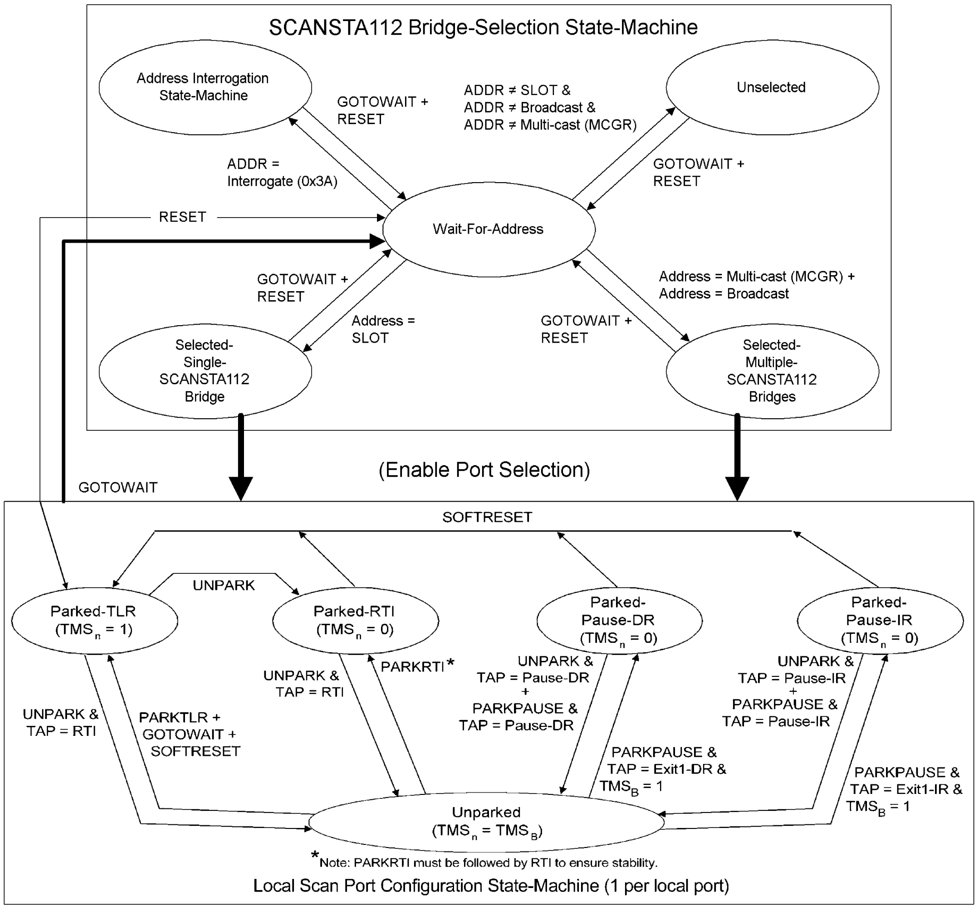
SCANSTA112 Bridge-Selection State-Machine
Address Interrogation State-Machine
GOTOWAIT + RESET
ADDR ≠ SLOT &
ADDR ≠ Broadcast &
ADDR ≠ Multi-cast (MCGR)
Unselected
ADDR = Interrogate (0x3A)
GOTOWAIT + RESET
RESET
Wait-For-Address
GOTOWAIT + RESET
Address = Multi-cast (MCGR) + Address = Broadcast
Address = SLOT
GOTOWAIT + RESET
Selected-Single-SCANSTA112 Bridge
Selected-Multiple-SCANSTA112 Bridges
(Enable Port Selection)
GOTOWAIT
SOFTRESET
UNPARK
Parked-TLR (TMS_n = 1)
Parked-RTI (TMS_n = 0)
Parked-Pause-DR (TMS_n = 0)
Parked-Pause-IR (TMS_n = 0)
PARKRTI*
UNPARK & TAP = RTI
UNPARK & TAP = Pause-DR + PARKPAUSE & TAP = Pause-DR
UNPARK & TAP = Pause-IR + PARKPAUSE & TAP = Pause-IR
UNPARK & TAP = RTI
PARKTLR + GOTOWAIT + SOFTRESET
PARKPAUSE & TAP = Exit1-DR & TMS_B = 1
PARKPAUSE & TAP = Exit1-IR & TMS_B = 1
Unparked (TMS_n = TMS_B)
*Note: PARKRTI must be followed by RTI to ensure stability.
Local Scan Port Configuration State-Machine (1 per local port)

Figure 11. Relationship Between SCANSTA112 State Machines

Following a reset, the SCANSTA112 TAP controller state machine is in the *Test-Logic-Reset* state; the SCANSTA112 selection state machine is in the *Wait-For-Address* state; and each of the seven local scan port configuration state machines is in the *Parked-TLR* state. The SCANSTA112 is then ready to receive Level 1 protocol instructions, *i.e.* an address, followed by Level 2 protocol instructions to select and unpark the desired LSPs.

The selection state machine and the local scan port configuration state machines implement the Level 1 and Level 2 protocols. The SCANSTA112 architecture includes a number of registers that control the operation of the SCANSTA112. These registers are accessed via the backplane TAP using the required and optional instructions described in the IEEE 1149.1 standard. We next describe the register set of the SCANSTA112.

8 Register Set

The SCANSTA112 includes a number of registers which are used for local port selection, SCANSTA112 configuration, scan data manipulation, and scan-support operations. These registers are listed in Table 6.

The bit fields and functions of each of these registers are detailed in Section 8.1.

Note that when any of these registers is selected for insertion into the active scan chain, scan data enters through the SCANSTA112 TDI_B input into that register's most-significant bit. Similarly, data that is shifted out of the register's least-significant bit is fed to the TDO_n output of the first selected LSP. The data then enters the TDI input of the next downstream device in the scan chain.

Table 6. Register Summary

Register Name	Bits	BSDL Name	Description
Instruction Register	8	INSTRUCTION	SCANSTA112 addressing and instruction-decode IEEE Std. 1149.1 required register
Boundary Register	22	BOUNDARY	IEEE Std. 1149.1 required register
Bypass Register	1	BYPASS	IEEE Std. 1149.1 required register
Device Identification Register	32	IDCODE	IEEE Std. 1149.1 optional register
Multi-Cast Group Register	2	MCGR	SCANSTA112 group address assignment
Mode Register 0	8	MODE0	SCANSTA112 local-port configuration and control bits
Mode Register 1	8	MODE1	SCANSTA112 local-port configuration and control bits
Mode Register 2	8	MODE2	SCANSTA112 Shared GPIO configuration bits
Shared GPIO Registers$_{(0-6)}$	8	SGPIOn	SCANSTA112 Shared GPIO control bits (one register per LSP)
Linear-Feedback Shift Register	16	LFSR	SCANSTA112 scan-data compaction (signature generation)
TCK Counter Register	32	CNTR	Local-port TCK clock-gating (for BIST)
Control Register	8	CONTROL	Control ScanBridge/stitcher mode, master port selection, and more
LSP Select Register	8	LSPSEL	Selection of LSPs for stitcher mode

8.1 *Register Descriptions*

The values in the registers listed in Table 6 control the operation of the SCANSTA112. These values are read and written over the backplane TAP as described in IEEE 1149.1. Many of these registers control multiple functions.

All the registers are shift registers. For all the registers, data shifted in on the TDI_B input enters the selected register at the most-significant bit position, propagates through the registers to the least-significant bit position, and is shifted out of the TDO output, which may be the TDO_B output or a LSP TDO output. We next describe the registers and their functions in detail.

Instruction Register: The instruction register is an 8-bit register that is in series with the scan chain whenever the TAP Controller of the SCANSTA112 is in the *Shift-Instruction-Register* state. This is one of the registers required by the IEEE 1149.1 standard.

In order to enter the *Shift-Instruction-Register* state the TAP controller must first pass through the *Capture-Instruction-Register* state. Upon entering the *Capture-Instruction-Register* state, the value XXXXXX01 is captured into the instruction register, where XXXXXX represents the value on the $S_{(0-5)}$ inputs. The IEEE 1149.1 standard specifies that in the *Capture-Instruction-Register* state the two least-significant bits of the instruction register must be loaded with 01. The other bits are not specified in the standard. The SCANSTA112 is compliant to the standard.

While the TAP controller is in the *Shift-Instruction-Register* state a new bit is shifted into the instruction register from the TDI_B input on each rising edge of the TCK_B line.

When the SCANSTA112 selection state machine is in the *Wait-For-Address* state, the instruction register is used for SCANSTA112 selection via address matching. To address an individual SCANSTA112, the selection state machine performs a comparison between a statically-configured (hardwired) value on the SCANSTA112's slot inputs and an address which is scanned into the SCANSTA112's instruction register.

Binary address codes 00000000 through 00111001 and 01000000 through 11111111 (00 through 39, Hex and 40 through FF Hex) are reserved for addressing individual SCANSTA112s. Addresses 3A Hex through 3F Hex are reserved for address interrogation and multiple addressing modes.

After the SCANSTA112 has been selected, the instruction register is inserted in the active scan chain whenever the TAP controller enters the *Shift-Instruction-Register* state. When the SCANSTA112 is selected, the instruction opcodes can be scanned into the instruction register. These instructions control the operation of the SCANSTA112. Instructions shifted into the instruction register are used to insert other registers in the SCANSTA112 into the scan chain for reading and writing.

Boundary Register: The boundary register is a sample-only register containing cells at all the inputs of the SCANSTA112. The register allows testing of circuitry and interconnects external to the SCANSTA112. This register is used for interconnect testing using the EXTEST instruction.

The boundary register for the SCANSTA112 includes input boundary cells on the following pins in this order: TLR_TRST_6, TLR_TRST, SB/$\overline{S}$, MPsel, TRANS, ADDMASK, LSPsel$_{(6-0)}$, $\overline{OE}$, S$_{(7-0)}$. All of these bits are inputs to the SCANSTA112; therefore the effect of the EXTEST instruction will be the same as that of the SAMPLE/PRELOAD instruction. Both instructions will capture the values on the input pins of the SCANSTA112 into the boundary register where they can be shifted out for boundary scan testing.

The boundary register is described in the BSDL file located on Texas Instrument's website. Note that cell 0 (the cell on static address pin S$_0$) is located closest to the TDO$_B$ output. Two BSDL files are provided for the SCANSTA112, one for LSP$_{B0}$ selected as the master backplane port and one for LSP$_{B1}$ selected as the master backplane port.

Bypass Register: The bypass register is a one-bit register that operates as specified in the IEEE 1149.1 standard once the SCANSTA112 has been selected. The register provides a minimum length serial path for shifting test data from the TDI$_B$ input and the first selected LSP.

This register is inserted in the scan chain by the BYPASS instruction. It is used when data from the TDI$_B$ input is intended for devices in the LSP rather than for a register in the SCANSTA112 itself. Insertion of the bypass register into the active scan chain shortens the serial access path to the data registers of other components in the active scan chain. When the bypass register is inserted in the scan chain, the data register path of the SCANSTA112 is only one bit long.

Device Identification Register: The device identification register is a 32-bit register compliant with IEEE Standard 1149.1. When the IDCODE instruction is scanned into the instruction register, the identification register is loaded with the predefined device identity code when the TAP controller next enters the *Capture-Data-Register* state. At this point it can be scanned out of the TDO port during the *Shift-Data-Register* state. Refer to our currently available BSDL file on the TI™ website for the device identity code for the current revision of the SCANSTA112.

Multi-Cast Group Register: Multi-cast addressing is a method of simultaneously selecting more than one SCANSTA112. The multi-cast group register (MCGR) is a 2-bit register used to assign a SCANSTA112 to one of four multi-cast groups. Four addresses are reserved for multi-cast addressing. When a SCANSTA112 in the *Wait-For-Address* state receives a multi-cast address, if that SCANSTA112's MCGR contains a value matching the group assigned to that multi-cast address, the SCANSTA112 becomes selected. Its selection state machine enters the *Selected-Multiple-SCANSTA112* state. The local scan port selection state machines become active and can be configured by Level 2 protocol instructions.

The MCGR is initialized to 00 upon entering the *Test-Logic-Reset* state. The mapping of the MCGR values to the multi-cast addresses is shown in Table 7.

Table 7. Multi-Cast Group Register Addressing

MCGR Bits 1,0	Hex Address	Binary Address
00	3C	00111100
01	3D	00111101
10	3E	00111110

Table 7. Multi-Cast Group Register Addressing (continued)

MCGR Bits 1,0	Hex Address	Binary Address
11	3F	00111111

The following boundary scan operations are used to perform multi-cast addressing.

1. Assign all target SCANSTA112s to a multi-cast group by writing each individual target SCANSTA112's MCGR with the same multi-cast group code (see Table 7). This configuration step must be done by individually addressing each target SCANSTA112 using its assigned slot value.
2. Scan the desired multi-cast group address into the TDI$_B$ input of all the SCANSTA112s. Note that this occurs in parallel, resulting in the selection of only those SCANSTA112s whose MCGR was previously programmed with the matching multi-cast group code.

Mode Register 0: Mode register 0 is an 8-bit data register used to select LSPs, to set the TCK configuration, to enable TDI$_B$ to TDO$_B$ loopback, and to report the status of the TCK counter. Mode register 0 is initialized to 00000001 binary when the TAP controller enters the *Test-Logic-Reset* state. Bits 0, 1, 2, and 4 are used for scan chain configuration as described in Table 10. When the UNPARK instruction is executed, the scan chain configuration will be as shown in Table 8 below. Note that the scan chain configuration depends upon the values of mode register 0 and mode register 1.

Table 8 shows the scan chain configuration when the selected LSPs are unparked. When all LSPs are parked, the scan chain configuration will be TDI$_B$ → SCANSTA112-register → TDO$_B$.

Table 8. Mode Register Control of LSP$_n$

Mode Register(s)	Scan Chain Configuration (if unparked)
MR0: X000X000MR1: XXXXX000	TDI$_B$ → Register → TDO$_B$
MR0: X000X001MR1: XXXXX000	TDI$_B$ → Register → LSP$_0$ → PAD → TDO$_B$
MR0: X000X010MR1: XXXXX000	TDI$_B$ → Register → LSP$_1$ → PAD → TDO$_B$
MR0: X000X011MR1: XXXXX000	TDI$_B$ → Register → LSP$_0$ → PAD → LSP$_1$ → PAD → TDO$_B$
MR0: X000X100MR1: XXXXX000	TDI$_B$ → Register → LSP$_2$ → PAD → TDO$_B$
MR0: X000X101MR1: XXXXX000	TDI$_B$ → Register → LSP$_0$ → PAD → LSP$_2$ → PAD → TDO$_B$
MR0: X000X110MR1: XXXXX000	TDI$_B$ → Register → LSP$_1$ → PAD → LSP$_2$ → PAD → TDO$_B$
MR0: X000X111MR1: XXXXX000	TDI$_B$ → Register → LSP$_0$ → PAD → LSP$_1$ → PAD → LSP$_2$ → PAD → TDO$_B$
MR0: X010X000MR1: XXXXX000	TDI$_B$ → Register → LSP$_3$ → PAD → TDO$_B$
MR0: X010X001MR1: XXXXX000	TDI$_B$ → Register → LSP$_0$ → PAD → LSP$_3$ → PAD → TDO$_B$
MR0: X010X010MR1: XXXXX000	TDI$_B$ → Register → LSP$_1$ → PAD → LSP$_3$ → PAD → TDO$_B$
MR0: X010X011MR1: XXXXX000	TDI$_B$ → Register → LSP$_0$ → PAD → LSP$_1$ → PAD → LSP$_3$ → PAD → TDO$_B$

Table 8. Mode Register Control of LSP_n (continued)

Mode Register(s)	Scan Chain Configuration (if unparked)
MR0: X010X100MR1: XXXXX000	TDI_B → Register → LSP_2 → PAD → LSP_3 → PAD → TDO_B
...	...
MR0: X110X111MR1: XXXXX000	TDI_B → Register → LSP_0 → PAD → LSP_1 → PAD → LSP_2 → PAD → LSP_3 → PAD → LSP_4 → PAD → TDO_B
MR0: X000X000MR1: XXXXX001	TDI_B → Register → LSP_5 → PAD → TDO_B
MR0: X000X001MR1: XXXXX001	TDI_B → Register → LSP_0 → PAD → LSP_5 → PAD → TDO_B
MR0: X000X010MR1: XXXXX001	TDI_B → Register → LSP_1 → PAD → LSP_5 → PAD → TDO_B
...	...
MR0: X110X111MR1: XXXXX001	TDI_B → Register → LSP_0 → PAD →→ LSP_1 → PAD → LSP_2 → PAD → LSP_3 → PAD → LSP_4 → PAD → LSP_5 → PAD → TDO_B
MR0: X000X000MR1: XXXXX010	TDI_B → Register → LSP_6 → PAD → TDO_B
...	...
MR0: X110X111MR1: XXXXX111	TDI_B → Register → LSP_0 → PAD → LSP_1 → PAD → LSP_2 → PAD → LSP_3 → PAD → LSP_4 → PAD → LSP_5 → PAD → LSP_6 → PAD → TDO_B
MR0: XXX1XXXXMR1: XXXXXXXX	TDI_B → Register → TDO_B (Loopback)

Bit 3 of mode register 0 is used for TCK_n configuration. The operation of the LSP TCK pins versus the settings of bit 3 of mode register 0 is shown in Table 9.

Bit 3 of mode register 0 is set to 0 by default so that TCK_n is free-running when the local scan ports are parked in the *Parked-RTI*, *Parked-Pause-DR* or *Parked-Pause-IR* state. When bit 3 is set to 1, the TCK_n for each LSP_n stops and is held low when LSP_n is parked. This feature can be used in power sensitive applications to reduce the power consumed by the boundary scan circuitry in parts of the system that are not being used for boundary scan operations. When LSP_n is in the *Parked-TLR* state, TCK_n is stopped after 512 clock pulses have been received on TCK_B independent of the bit 3 value. This insures that a hierarchical system consisting of SCANSTA112s on the LSPs of other SCANSTA112s will be reset when the LSPs of the SCANSTA112s at the first level of the hierarchy are placed in the *Parked-TLR* state.

Table 9. Test Clock Configuration

Bit 3 of Mode Register 0	LSP_n	TCK_n
1	Parked	Stopped
0	Parked	Free-running
1	Unparked	Free-running
0	Unparked	Free-running
X	Parked-TLR	Stopped after 512 clock pulses

Bit 7 of mode register 0 is a status bit for the TCK counter. Bit 7 is set to 1 when the TCK counter is on and has reached terminal count (zero). It is cleared (set to 0) when the counter is loaded following a CNTRSEL instruction. The power-on value for bit 7 is 0.

Mode Register 1: Mode register 1 is an 8-bit register which is used to select LSPs along with mode register 0. Bits 0 and 1 function similarly to mode register 0, but apply as shown in Table 10.

Bit 7 of mode register 1 is used to enable transparent LSP mode. Transparent LSP mode is activated when this bit is set to 1. On power up or reset, this bit is set to 0 by default. This mode is only available in ScanBridge primary operating mode.

Mode Register 2: The 8-bit mode register 2 is described in Table 10. This register selects which of the LSPs will be put in the shared GPIO mode. LSPs configured as shared GPIO ports by the settings of mode register 2 are controlled by the associated $SGPIO_n$ register.

A summary of the effects of the bits in all three mode registers is shown in Table 10.

Shared $GPIO_n$ Register: There are seven shared GPIO registers, one for each LSP. Each 8-bit register controls the function of the LSP TMS, TDO, and TDI pins as shown in Table 11 when the mode register 2 setting activates the corresponding LSP as a GPIO port.

Table 10. Mode Register Bits [1]

Bit	Mode Register 0	Mode Register 1	Mode Register 2
0	LSP_0	LSP_5	$LSP_0/GPIO_0$
1	LSP_1	LSP_6	$LSP_1/GPIO_1$
2	LSP_2	Reserved	$LSP_2/GPIO_2$
3	TCK Free Run Disable (Table 9)	Reserved	$LSP_3/GPIO_3$
4	TDI_B to TDO_B Loopback	Reserved	$LSP_4/GPIO_4$
5	LSP_3	Reserved	$LSP_5/GPIO_5$
6	LSP_4	Reserved	$LSP_6/GPIO_6$
7	TCK Counter Status	Enable Transparent LSP Mode	Reserved

[1] The default value for all of the mode register bits is 0 except for bit 0 of mode register 0 which is 1

Table 11. Shared GPIO Register Bits [1]

Bit	SGPIO Register n
0	Output (TMS)
1	Output (TDO)
2	Input (TDI)
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved

[1] The default value for all of the shared GPIO register bits is 0.

Linear Feedback Shift Register: The SCANSTA112 contains a signature compactor which supports test result evaluation when multiple SCANSTA112s are selected. The signature compactor consists of a 16-bit linear feedback shift register (LFSR) which can monitor LSP data as it is shifted into the LSP TDO line from the SCANSTA112's local scan chain. Once the LFSR is enabled, the LFSR's state changes in a reproducible way as each LSP data bit is shifted in from the local scan chain. When all local scan chain data has been scanned in, the LFSR contains a 16-bit signature value which can be compared against an expected signature computed for the local scan chain.

The LFSR uses the following feedback polynomial:

$F(x) = X^{16} + X^{12} + X^3 + X + 1$

The LFSR allows users to test long scan chains in parallel, via broadcast or multi-cast addressing modes, and check only the 16-bit signatures from each SCANSTA112. The LFSR is initialized with a value of 0000 Hex upon reset. The LFSRSEL instruction can be used to select the LFSR for scanning out a 16-bit signature result or for scanning in a 16-bit seed.

TCK Counter Register: The 32-bit TCK counter register is used for BIST testing as defined in the IEEE 1149.1 standard. The standard does not require that BIST be implemented in a boundary scan-enabled device. If BIST is implemented, however, the standard requires that BIST be implemented in a specific way.

To run BIST on a device, a specified number of TCK cycles are input to the device under test while its TAP controller is in the *Run-Test/Idle* state. This is what the TCK counter register is designed to implement. Using the TCK counter register, BIST can be run on a parked LSP while another SCANSTA112 LSP is being used for boundary scan operations.

The CNTRSEL instruction is used to load a initial value into the counter register via the backplane TAP. The counter is enabled by the CNTRON instruction. When the LSPs which are to perform BIST are parked, the TCK for each parked LSP will continue to run to perform the BIST. Each LSP TCK pulse will decrement the value in the TCK counter register until the TCK counter register reaches zero. The LSP TCKs will then stop and be held low. When the TCK register reaches its terminal count (00000000 Hex), bit 7 of mode register 0 will be set to 1.

If a readout of the BIST result is desired, each LSP can be unparked and its BIST result scanned out. The register that captures the BIST result and the expected values of that register are not specified by the IEEE 1149.1 standard but are defined for each boundary scan-enabled device in the system by its manufacturer.

The TCK counter is initialized with a value of 00000000 Hex upon reset.

Control Register: The control register is an 8-bit register as shown in Table 12. All bits are initialized with default values when the SCANSTA112 is reset. Five of the bits are initialized by values on external pins. Bit 0, the IgnoreReset bit, has a default value of 0. See Table 2 and Table 3 for information on resetting the SCANSTA112.

The user can overwrite the default contents of this register by issuing the CONTROLSELECT Instruction. After scanning in this instruction, when the backplane TAP sequences through the *Update-Instruction-Register* state the control register will be inserted into the active scan chain to be written by data shifted in when the TAP controller enters the *Shift-Data-Register* state.

Writing new values to the control register can place the SCANSTA112 in an undesired state. Changing the value of the MPsel bit, for example, will result in changing the selected master port. Since the boundary scan controller is probably connected to only one of the two available backplane TAPs, it will no longer be possible to control the SCANSTA112 over the backplane TAP. Also, The TMS_B, TCK_B, and $\overline{\text{TRST}}_{\overline{B}}$ pins connected to the boundary scan controller will change direction (pins that were inputs will become outputs, and vice-versa), which will result in bus contention. The value of this bit should be controlled by the use of the external $\text{MPsel}_{B1/\overline{B0}}$ pin rather than by writing directly to the control register. Changing the MPsel bit in the control register without resetting the SCANSTA112 is an undefined operation for this reason.

The functions of most of the other bits in this register have been previously discussed. The effect of the IgnoreReset bit is shown in Table 2 and Table 3. The SB/$\overline{\text{S}}$ bit selects ScanBridge or stitcher mode. The TRANS bit sets the SCANSTA112 to full transparent scan chain mode when it is in stitcher primary mode.

Bits 4 and 5 of the control register, the TLR_TRST bit and the TLR_TRST_6 bit, control the value presented on the LSP $\overline{\text{TRST}}$ pins when a LSP is parked in the *Test-Logic-Reset* state.

When a SCANSTA112 LSP is parked in *Test-Logic-Reset*, all the pins of the LSP are driven to safe values. Normally, for example, $\overline{\text{TRST}}_{\overline{n}}$ is driven to 0. The value of 0 for $\overline{\text{TRST}}_{\overline{n}}$ was selected to insure that all TAPs that are connected to LSPs are held in *Test-Logic-Reset*. However, there are devices which do not function properly when their $\overline{\text{TRST}}$ pin is held low. To accommodate these devices, the SCANSTA112 has two additional input pins, TLR_TRST and TLR_TRST_6. The values on these pins will be the default values driven on the $\overline{\text{TRST}}_{\overline{(0-5)}}$ and $\overline{\text{TRST}}_{\overline{6}}$ pins respectively while the TAP controllers for the associated LSPs are in the *Test-Logic-Reset* state.

Regardless of the SCANSTA112 primary mode of operation and the value of the IgnoreReset bit, the $\overline{\text{TRST}}$ pins of all LSPs, parked or unparked, will be forced to 0 when the $\overline{\text{TRST}}_{\overline{B}}$ pin is actively driven low by the boundary scan controller. The TLR_TRST and TLR_TRST_6 pins determine the value of $\overline{\text{TRST}}$ for a LSP when the LSP is parked in *Test-Logic-Reset* state but the backplane reset is not active. If the TLR_TRST pin is set to 1, the outputs of the $\overline{\text{TRST}}_{\overline{(0–5)}}$ pins will force a 1 when the LSPs are parked in *Test-Logic-Reset* state except when $\overline{\text{RESET}}$ or $\overline{\text{TRST}}_{\overline{B}}$ is forced by the boundary scan controller to a 0. The TLR_TRST_6 pin controls the state of the $\overline{\text{TRST}}_{\overline{6}}$ in the same way.

The intent is that devices that operate properly in *Test-Logic-Reset* state with their $\overline{\text{TRST}}$ inputs held low will be apportioned to LSPs 0–5 while devices which require their $\overline{\text{TRST}}$ to be held high for proper operation in *Test-Logic-Reset* state will be apportioned to LSP 6. The TLR_TRST and TLR_TRST_6 pins can then be used to set the appropriate values of the $\overline{\text{TRST}}$ pin for each LSP. Other combinations are possible as well.

Table 12. Control Register

Bit	Function	Default Value
0	IgnoreReset	0
1	TRANS	External Pin
2	SB/$\overline{\text{S}}$	External Pin
3	MPsel$_{B1/\overline{B0}}$	External Pin
4	TLR_TRST	External Pin
5	TLR_TRST_6	External Pin
6	Future Expansion	0
7	Future Expansion	0

LSP Select Register: The LSP select register is an 8-bit register. The most-significant bit is reserved and set by default to 0. Bits 6–0 of this register are initialized to default values when the SCANSTA112 is powered up or reset. The default values of these bits are set based on the value of the LSPsel$_{(0–6)}$ pins.

The contents of this register determines which of the LSPs are to be stitched together in stitcher mode. If LSPsel$_n$ is set to 1 then LSP$_n$ will be included in the scan chain. The value of this register can be modified allowing the scan chain configuration to be modified. In ScanBridge mode the selected LSPs are determined by the settings of mode registers 0–2. In ScanBridge mode the LSP select register has no effect on the LSP configuration.

We next describe the instructions available in the SCANSTA112 instruction set and their effects on the operation of the SCANSTA112.

9 SCANSTA112 Instruction Set

Once the SCANSTA112 is addressed, its operation is controlled by instructions scanned into its instruction register from the backplane TAP. Some instructions are used to place one of the SCANSTA112 registers into the scan chain so that its value may be written or read out. Others effect immediate changes in the operation of the SCANSTA112 without changing the contents of the device registers. The Level 2 protocol instructions described previously fall into this category.

Once the SCANSTA112 has been selected, it is ready to receive instructions over the backplane TAP. At this point the internal registers of the SCANSTA112 may be accessed by scanning the appropriate instructions into the TDI$_B$ input.

Upon being selected the SCANSTA112 selection controller state machine transitions from the *Wait-For-Address* state to one of the selected states. Each of the LSPs remains parked in one of the following four TAP Controller states: *Test-Logic-Reset*, *Run-Test/Idle*, *Pause-Data-Register*, or *Pause-Instruction-Register*. In this condition the active scan chain will be configured as follows.

TDI$_B$ → Instruction Register → TDO$_B$

The UNPARK instruction is used to insert one or more LSPs into the active scan chain. Table 8 describes which local ports are inserted into the chain for various values in mode register 0 and mode register 1, and in what order they are inserted.

The SCANSTA112 instruction set provides a number of instructions for functions other than LSP configuration. These instructions provide access to the various registers within the SCANSTA112 and control of the SCANSTA112.

9.1 SCANSTA112 Instruction Types

The SCANSTA112 instruction set includes two basic types of instructions.

1. Instructions that insert a SCANSTA112 register into the active scan chain so that the register can be captured or updated. These instructions include BYPASS, SAMPLE/PRELOAD, EXTEST, IDCODE, MODESEL, MODESEL1, MODESEL2, MCGRSEL, LFSRSEL, CNTRSEL, CONTROLSEL, and LSPSEL. These instructions cause a target register to be inserted into the active scan chain.
2. Instructions that configure local ports or control the operation of the linear feedback shift register and counter registers. The Level 2 protocol instructions described previously are included in this category. These instructions include UNPARK, PARKTLR, PARKRTI, PARKPAUSE, GOTOWAIT, SOFTRESET, LFSRON, LFSROFF, CNTRON, CNTROFF, TRANSPARENTENABLE and TRANSPARENTn. These instructions, along with any undefined opcodes, cause the device identification register to be inserted into the active scan chain.

The instructions available in the SCANSTA112 instruction set, their associated opcodes, and the target registers they cause to be inserted in the active scan chain are summarized in Table 13.

Table 13. SCANSTA112 Instructions, Opcodes, and Target Registers

Instructions	Hex Opcode	Binary Opcode	Data Register
BYPASS	FF	1111 1111	Bypass Register
EXTEST	00	0000 0000	Boundary Register
SAMPLE/PRELOAD	81	1000 0001	Boundary Register
IDCODE	AA	1010 1010	Device Identification Register
UNPARK	E7	1110 0111	Device Identification Register
PARKTLR	C5	1100 0101	Device Identification Register
PARKRTI	84	1000 0100	Device Identification Register
PARKPAUSE	C6	1100 0110	Device Identification Register
GOTOWAIT [(1)]	C3	1100 0011	Device Identification Register
MODESEL	8E	1000 1110	Mode Register 0
MODESEL1	82	1000 0010	Mode Register 1
MODESEL2	83	1000 0011	Mode Register 2
MCGRSEL	03	0000 0011	Multi-Cast Group Register
SOFTRESET	88	1000 1000	Device Identification Register
LFSRSEL	C9	1100 1001	Linear Feedback Shift Register
LFSRON	0C	0000 1100	Device Identification Register
LFSROFF	8D	1000 1101	Device Identification Register
CNTRSEL	CE	1100 1110	32-Bit TCK Counter Register
CNTRON	0F	0000 1111	Device Identification Register
CNTROFF	90	1001 0000	Device Identification Register
DEFAULT_BYPASS	07	0000 0111	Set the bypass register as the default data register
TRANSPARENT0	A0	1010 0000	(no register - Transparent to LSP_0)
TRANSPARENT1	A1	1010 0001	(no register - Transparent to LSP_1)
TRANSPARENT2	A2	1010 0010	(no register - Transparent to LSP_2)
TRANSPARENT3	A3	1010 0011	(no register - Transparent to LSP_3)
TRANSPARENT4	A4	1010 0100	(no register - Transparent to LSP_4)
TRANSPARENT5	A5	1010 0101	(no register - Transparent to LSP_5)
TRANSPARENT6	A6	1010 0110	(no register - Transparent to LSP_6)
SGPIO0	B8	1011 1000	Shared GPIO $Register_0$

[(1)] All instructions other than GOTOWAIT act on selected SCANSTA112s only. The GOTOWAIT instruction acts on all SCANSTA112s in the system whether they are selected or not.

Table 13. SCANSTA112 Instructions, Opcodes, and Target Registers (continued)

Instructions	Hex Opcode	Binary Opcode	Data Register
SGPIO1	B9	1011 1001	Shared GPIO Register$_1$
SGPIO2	BA	1011 1010	Shared GPIO Register$_2$
SGPIO3	BB	1011 1011	Shared GPIO Register$_3$
SGPIO4	BC	1011 1100	Shared GPIO Register$_4$
SGPIO5	BD	1011 1101	Shared GPIO Register$_5$
SGPIO6	BE	1011 1110	Shared GPIO Register$_6$
CONTROLSEL	87	1000 0111	Control Select Register
LSPSEL	86	1000 0110	LSP Select Register
TRANSPARENTENABLE	A8	1010 1000	(no register - Transparent to selected LSPs)
TESTMODE	37	0011 0111	Used for factory test mode only
Other Undefined	TBD	TBD	Device Identification Register

9.2 Instruction Descriptions

The effects of each instruction in Table 13 are described in this section. Some of this information has been presented in previous sections but is repeated here for easier reference.

BYPASS: The BYPASS instruction selects the bypass register for insertion into the active scan chain when the SCANSTA112 is selected.

EXTEST: The EXTEST instruction selects the boundary register of the SCANSTA112 for insertion into the active scan chain. The boundary register consists of sample-only (input) boundary cells connected to the SCANSTA112 pins as described in the SCANSTA112 BSDL file. The BSDL file can be found on TI's website. Note that cell 0 (S_0) is located closest to TDO_B. On the SCANSTA112, the EXTEST instruction performs the same function as the SAMPLE/PRELOAD instruction, since all the boundary cells in the SCANSTA112 are input boundary cells.

SAMPLE/PRELOAD: The SAMPLE/PRELOAD instruction selects the boundary register for insertion into the active scan chain. The boundary register consists of sample-only boundary cells connected to the SCANSTA112 pins as described in the SCANSTA112 BSDL file. The BSDL file can be found on TI's website. Note that cell 0 (S_0) is located closest to TDO_B. Since the SCANSTA112 boundary register consists only of input cells the SAMPLE/PRELOAD instruction has the same effect as the EXTEST instruction.

IDCODE: The IDCODE instruction selects the device identification register for insertion into the active scan chain. When IDCODE is the current active instruction the device identification value is captured into the device identification register when the TAP controller enters the *Capture-Data-Register* state.

UNPARK: This instruction unparks the selected LSPs and inserts them into the active scan chain. The LSPs are selected by the values of mode register 0 and mode register 1 in ScanBridge mode or by the value of the LSP select register in stitcher mode.

The TAP controllers of all the unparked LSPs are sequenced synchronously with the SCANSTA112's TAP controller. When an LSP has been parked in the *Run-Test/Idle* state, it will become unparked when the SCANSTA112's TAP Controller enters the *Run-Test/Idle* state after the UNPARK instruction is scanned into the SCANSTA112 instruction register.

An LSP which has been parked in *Test-Logic-Reset* passes through the *Run-Test/Idle* state prior to being unparked. The LSP will be parked in the *Run-Test/Idle* state upon update of an UNPARK instruction. The LSP will be unparked when the TAP controller is sequenced through the*Run-Test/Idle* state .

If an LSP has been parked in one of the stable pause states, *Pause-Data-Register* or *Pause-Instruction-Register*, it will become unparked when the SCANSTA112's TAP Controller enters the stable state in which it is paused.

PARKTLR: This instruction causes all unparked LSPs to be parked in the *Test-Logic-Reset* state and removes the LSPs from the active scan chain. The LSP controllers keep the LSPs parked in the *Test-Logic-Reset* state by forcing their respective TMS_n outputs with a constant 1 as long as the local scan port configuration state machine is in the *Parked-TLR* state. In stitcher mode the LSPs pass through the *Test-Logic-Reset* state, but are really finally parked in the *Run-Test/Idle* state rather than the *Test-Logic-Reset* state.

PARKRTI: This instruction causes all unparked LSPs to be parked in the *Run-Test/Idle* state. The update of the PARKRTI instruction must immediately be followed by the boundary scan controller driving the TMS_B line to 0 which causes the backplane TAP controller to enter the *Run-Test/Idle* state. This is similar to the sequence of operations required when a PARKPAUSE instruction is issued. This is described below.

When a LSP_n is unparked, its TMS_n signals follow TMS_B and the LSP_n TAP controller state transitions are synchronized with the TAP controller state transitions of the SCANSTA112. When the instruction register is updated with the PARKRTI instruction, TMS_n will be forced to a constant logic 0, causing the unparked local TAP controllers to be parked in the *Run-Test/Idle* state. When a LSP is parked, it is removed from the active scan chain.

PARKPAUSE: The PARKPAUSE instruction has dual functionality. It can be used to park unparked LSPs or to unpark parked LSPs.

If the SCANSTA112 receives the PARKPAUSE instruction when it has unparked LSPs, the instruction will ready all the unparked LSPs for parking in one of the TAP controller pause states. The unparked LSPs will not be parked until the SCANSTA112's TAP Controller is sequenced through *Exit1-Data-Register* or *Exit1-Instruction-Register* state into the *Update-Data-Register* or *Update-Instruction-Register* state **without** passing through the *Pause-Data-Register* or *Pause-Instruction-Register* state. Passing through the *Pause-Data-Register* or *Pause-Instruction-Register* state would unpark the LSPs.

When the SCANSTA112 TAP Controller is in the *Exit1-Data-Register* or *Exit1-Instruction-Register* state and TMS_B is high, the LSP controller forces a constant logic 0 onto the LSP TMS lines. This will park the LSPs in the *Pause-Data-Register* or *Pause-Instruction-Register* state. At this point, with the boundary scan controller driving a 1 on the TMS_B line, the SCANSTA112 TAP controller will go to the *Update-Data-Register* or *Update-Instruction-Register* state. The LSPs, however, will transition to the *Pause-Data-Register* or *Pause-Instruction-Register* state.

The PARKPAUSE instruction is implemented with this dual functionality to enable backplane testing (interconnect testing between boards) with simultaneous updates and captures.

After the LSPs have been parked, another instruction can be scanned into the SCANSTA112 instruction register to reconfigure the local ports or to deselect the SCANSTA112 (*e.g.* MODESEL, GOTOWAIT, etc.). The LSPs will remain parked in the *Pause-Data-Register* or *Pause-Instruction-Register* state.

Consider the state of the SCANSTA112 and its LSPs after the PARKPAUSE instruction has been received and the LSPs have been parked in one of the pause states. The active instruction in the SCANSTA112 is still the PARKPAUSE instruction, but a new instruction has been shifted into the instruction register. The new instruction has not yet been captured, so it is not yet active. If the TAP controller is sequenced through the *Exit1-Instruction-Register* and *Update-Instruction-Register* states without passing through the *Pause-Instruction-Register* state, the LSPs will remain parked. Once the TAP controller passes through the *Update-Instruction-Register* state the active instruction will no longer be PARKPAUSE, so the LSPs will remain parked until the PARKPAUSE or UNPARK instruction is scanned in.

If the PARKPAUSE instruction is given to a SCANSTA112 whose LSPs are parked in *Pause-Instruction-Register* or *Pause-Data-Register*, the parked LSPs will become unparked when the SCANSTA112's TAP controller is sequenced into the respective pause state. The UNPARK instruction has the same effect on parked LSPs.

This behavior of the PARKPAUSE instruction permits efficient interconnect testing using multiple SCANSTA112s. This is accomplished as follows.

1. Reset all the SCANSTA112s in the system. This puts all the SCANSTA112s in the *Wait-for-Address* state.
2. Address the first SCANSTA112 using its unique slot address. This SCANSTA112 goes to the *Selected-Single-SCANSTA112* state. All the other SCANSTA112s in the system go to the *Unselected*

state.

3. Configure the LSPs of the first SCANSTA112 and unpark them.
4. Issue the PRELOAD instruction to all the devices on the LSPs and the BYPASS instruction to the SCANSTA112.
5. Load the desired boundary scan data into the boundary registers of the LSPs. This data will consist of the test values to be driven on the output pins of the boundary scan-enabled devices on the LSPs of the selected SCANSTA112.
6. Issue the EXTEST instruction to the devices on the selected SCANSTA112's LSPs and the PARKPAUSE instruction to the selected SCANSTA112. When the TAP controller sequences through the *Exit1-Instruction-Register* to the *Update-Instruction-Register* state the LSPs will be parked in the *Pause-Instruction-Register* state.
7. Issue the GOTOWAIT instruction. Make sure that the TAP controller does not pass through the *Pause-Instruction-Register* state which would unpark the LSPs. The selected SCANSTA112 will go to the *Wait-for-Address* state. All the other SCANSTA112s will already be in the *Wait-for-Address* state. The LSPs of the previously-selected SCANSTA112 will remain parked in the *Pause-Instruction-Register* state.
8. Address each SCANSTA112 in turn and load the desired boundary scan data into the boundary registers of the devices on its LSPs. At the end of this sequence of instructions: all the SCANSTA112s will be in the *Wait-for-Address* state; all the LSPs of these SCANSTA112s will be parked in the *Pause-Instruction-Register* state; the boundary scan data will be loaded into the boundary registers of all the devices on all the LSPs; all the devices on all the LSPs will have received the EXTEST instruction; and all the devices on all the LSPs will be ready to execute the EXTEST instruction.
9. Broadcast address all the desired SCANSTA112s.
10. Issue all the SCANSTA112s the PARKPAUSE instruction.
11. Cycle the TAP controller through *Capture-Instruction-Register*, *Exit1-Instruction-Register*, *Pause-Instruction-Register* (unparking all the LSPs), *Exit2-Instruction-Register*, *Update-Instruction-Register* and *Run-Test/Idle* states. This will unpark all the LSPs.
12. Cycle the TAP controller through the *Capture-Data-Register* (which executes the EXTEST instruction), *Exit1-Data-Register* (which parks all the LSPs again), *Update-Data-Register*, and *Run-Test/Idle* states. This will perform the boundary scan test set up by the EXTEST instruction. This will capture the data on each device's pins in its boundary register. Since the PARKPAUSE instruction is still active in the SCANSTA112s, all the LSPs will now be parked in the *Pause-Data-Register* state.
13. Scan the GOTOWAIT instruction into all the SCANSTA112s. This will send them all into the *Wait-for-Address* state. All the LSPs will still be parked in the *Pause-Data-Register* state.
14. Address the first SCANSTA112 directly. Send it the PARKPAUSE instruction.
15. Cycle the backplane TAP controller through the *Select-Data-Register-Scan*, *Capture-Data-Register*, *Exit1-Data-Register*, *Pause-Data-Register* (unparking the LSPs of the selected SCANSTA112), *Exit2-Data-Register*, and back to the *Shift-Data-Register* state. Since the LSPs do not go through the *Capture-Data-Register* state (they are parked when the TAP controller enters this state) the contents of the boundary registers retain their values from the previous EXTEST instruction and can be shifted out. The EXTEST instruction is still active in each device on each LSP. The PARKPAUSE instruction is still active in the SCANSTA112.
16. Scan out the results of the previous test from the boundary register and scan in the values for the next test.
17. After scanning the values for the next test into the boundary register, cycle the TAP controller through the *Exit1-Data-Register*, *Update-Data-Register*, and *Run-Test/Idle* state. This will park the LSPs of the selected SCANSTA112 in the *Pause-Data-Register* state.
18. Scan the GOTOWAIT instruction into all the SCANSTA112s. This will send them all into the *Wait-for-Address* state. All the LSPs will still be parked in the *Pause-Data-Register* state.
19. Address each SCANSTA112 in turn. For each SCANSTA112, scan out the current boundary scan result and load the desired boundary scan data into the boundary registers of the devices on its LSPs. At the end of this sequence of instructions: all the SCANSTA112s will be in the *Wait-for-Address* state; all the LSPs of these SCANSTA112s will be parked in the *Pause-Data-Register* state; the boundary scan data will be loaded into the boundary registers of all the devices on all the LSPs; and all the devices will be ready to execute the EXTEST instruction again.

20. Broadcast address all the desired SCANSTA112s.
21. Issue all the SCANSTA112s the PARKPAUSE instruction.
22. Cycle the TAP controller through *Capture-Data-Register*, *Exit1-Data-Register*, *Pause-Data-Register* (unparking all the LSPs), *Exit2-Data-Register*, *Update-Data-Register* and *Run-Test/Idle* states. This will unpark all the LSPs
23. Cycle the TAP controller through the *Capture-Data-Register* (which executes the EXTEST instruction again), *Exit1-Data-Register* (which parks all the LSPs again), *Update-Data-Register*, and *Run-Test/Idle* states. This will perform the boundary scan test set up by the EXTEST instruction. This will capture the data on each device's pins in its boundary register. Since the PARKPAUSE instruction is still active in the SCANSTA112s, all the LSPs will again be parked in the *Pause-Data-Register* state.
24. Repeat the above sequence for as many test vectors as desired.
25. Reset all the SCANSTA112s in the system. This puts all the SCANSTA112s in the *Wait-for-Address* state.

The dual function of the PARKPAUSE instruction allows the system controller to perform the EXTEST function using the broadcast address for the SCANSTA112 and then scan out the data for each SCANSTA112 without executing the EXTEST instruction again. This enables boundary scan testing between LSPs on different SCANSTA112s, which may be on different PC boards, using the broadcast address.

GOTOWAIT: This instruction is used to return all SCANSTA112s to the *Wait-For-Address* state. All unparked LSPs will be parked in the *Test-Logic-Reset* state (see Figure 11). This is the only instruction that all SCANSTA112s will respond to whether they have been addressed or not. The SCANSTA112 will not respond to the GOTOWAIT instruction if it is in full transparent scan chain mode.

MODESELn: The MODESELn instruction inserts mode register n into the active scan chain. The MODESEL instruction inserts mode register 0 into the active scan chain, the MODESEL1 inserts mode register 1, and so forth.

MCGRSEL: This instruction inserts the multicast group register (MCGR) into the active scan chain. The MCGR is used to group SCANSTA112s into multi-cast groups for parallel TAP sequencing (*i.e.*, to simultaneously perform identical scan operations).

SOFTRESET: This instruction causes all seven port configuration controllers (see Figure 11) to enter the *Parked-TLR* state, which forces TMS_n high; this parks each local port in the *Test-Logic-Reset* state within five TCK_B cycles. A LSP parked in the *Test-Logic-Reset* state can be unparked by issuing the UNPARK instruction and then sequencing the TAP controller through the *Run-Test/Idle* state.

LFSRSEL: This instruction inserts the linear feedback shift register (LFSR) into the active scan chain, allowing a compacted signature to be shifted out of the LFSR during the *Shift-Data-Register* state. (The signature is assumed to have been computed during earlier LFSRON shift operations.) This instruction disables the LFSR register's feedback circuitry, turning the LFSR into a standard 16-bit shift register. This instruction allows a signature to be shifted out of the LFSR or a seed value to be shifted into it.

LFSRON: Once this instruction is executed, the linear feedback shift register samples data from the active scan path (including all unparked LSPs) while the TAP controller is in the *Shift-Data-Register* state. Data from the scan path is shifted into the linear feedback shift register and compacted. This allows a serial stream of data to be compressed into a 16-bit signature that can subsequently be shifted out using the LFSRSEL instruction. The linear feedback shift register is not placed in the scan chain during this mode. Instead, the register samples the active scan-chain data as it flows from the last unparked LSP_n to TDO_B. The LFSR is often used when a group of SCANSTA112s is addressed in broadcast or multi-cast addressing modes. In this application the LFSR accumulates a signature from the scan chain which will be shifted out later when each SCANSTA112 is individually addressed and its TDO_B line is no longer at TRI-STATE.

LFSROFF: This instruction terminates linear feedback shift register sampling. The LFSR retains its current state after receiving this instruction.

CNTRSEL: This instruction inserts the 32-bit TCK counter shift register into the active scan chain. This allows the user to program the number of n TCK cycles to send to the parked local ports once the CNTRON instruction is issued (*e.g.*, for BIST operations). Note that to ensure completion of the countdown, the SCANSTA112 should receive at least n TCK_B pulses after the LSPs are parked.

CNTRON: This instruction enables the TCK counter. The counter begins counting down on the first rising edge of TCK_B following the *Update-Instruction-Register* TAP controller state and is decremented on each rising edge of TCK_B thereafter. When the TCK counter reaches terminal count, 00000000 Hex, TCK_n of all parked LSPs are held low. This function overrides the mode register 0 TCK control bit (bit 3).

If the CNTRON instruction is issued when the TCK counter is 00000000 (terminal count) the local TCKs of parked LSPs will be gated (driven to a static 0). The counter will begin counting on the rising edge of TCK_B when the TCK counter is loaded with a non-zero value following a CNTRSEL instruction (see Section 10.3 for an example).

CNTROFF: This instruction disables the TCK counter, and TCK_n control is returned to mode register 0 (bit 3).

DEFAULT_BYPASS: This instruction selects the Bypass register to be the default for SCANSTA112 commands that do not explicitly require a data register. The default register for such operations after RESET is the Device ID register. This command changes that default behavior.

CONTROLSEL: This instruction inserts the control register into the active scan chain. Caution should be used when writing to this register. Changing the value of the MPsel bit , for example, will result in the reconfiguration of the backplane master port. When this occurs the boundary scan controller, which is presumably connected to only one of the SCANSTA112 backplane master ports, will lose control of the SCANSTA112. Also, the TMS_B, TCK_B, and $\overline{TRST}_{\overline{B}}$ pins connected to the boundary scan controller change directions (input pins become outputs and vice-versa), and this will result in bus contention.

LSPSEL: This instruction inserts the LSP select register into the active scan chain.

TRANSPARENTENABLE: This instruction is only available when the SCANSTA112 is in ScanBridge mode. It is used along with the contents of mode register 0 and mode register 1 to place the selected LSPs into the scan chain without SCANSTA112 registers or pad bits (full transparent mode).

TRANSPARENTn: These 7 instructions are only available when the SCANSTA112 is in ScanBridge mode. They are used to select the a single LSP, LSP_n, and place it alone into full transparent mode. This removes all SCANSTA112 registers and pad bits from the scan chain and inserts LSP_n. There are no registers associated with this set of instructions as the SCANSTA112 becomes transparent when it receives this instruction.

TESTMODE: This Instruction places the SCANSTA112 into production scan test-mode. This instruction is reserved and should not be used in system.

10 Additional Features

The SCANSTA112 provides extended capability in a boundary scan system. The SCANSTA112 is designed to provide this extended capability in an intuitive way by interfacing smoothly to a system implementing the IEEE 1149.1 standard.

In this section we provide some additional information about the extended functionality available with the SCANSTA112. Some of the features discussed here have been mentioned previously. In this section we provide a more in-depth treatment of the additional features of the SCANSTA112.

10.1 Master Port Selection

The SCANSTA112 can be configured so that either of two available master ports, either the backplane B_0 port, or the backplane B_1 port, is the backplane master port. This feature is useful to allow an alternate boundary scan controller to take control of the scan chains.

Selection of the master port is accomplished by using the master port select pin, $MPsel_{B1/\overline{B0}}$. When the SCANSTA112 is reset the value on this pin is loaded into the MPsel bit of the control register.

To accommodate the change in position of the master port boundary scan cells, there are two BSDL files available on TI's website, one to support each mode.

Changing the state of the $MPsel_{B1/\overline{B0}}$ input without a reset is an undefined operation. Selection of the master port should always be accomplished by setting the value of the master port select pin, $MPsel_{B0/\overline{B1}}$, and resetting the SCANSTA112.

The TAP port which is not selected as the master port will assume the function of LSP_{00}. Some additional explanation of the mapping of the master backplane ports is provided in Section 10.9.

10.2 LSP Output Enable

The LSP pins can be at TRI-STATE by setting the value of the $\overline{OE}$ pin to 1 (high). Normally this pin should be tied low. If it is driven high, all the outputs of all the LSPs will be at TRI-STATE. In this mode, these pins can be safely driven from an alternate source. This provides additional flexibility for the system designer.

10.3 BIST Support

Many JTAG-enabled devices implement the RUNBIST instruction in some form. The IEEE 1149.1 standard does not require this instruction and does not specify the opcode to be used for it if it is implemented.

The standard does specify that the BIST is to performed by loading the RUNBIST instruction into the target device and sequencing its TAP controller to the *Run-Test/Idle* state. While in the *Run-Test/Idle* state the BIST sequence runs. The BIST sequence may require a specified number of TCK pulses or may require that the TAP controller remain in the *Run-Test/Idle* state for a specified duration. When the BIST sequence is complete a specified target register is to be loaded with the BIST result so that the boundary scan controller can scan this result out and compare it to the expected BIST result. The RUNBIST_EXECUTION attribute in the BSDL file of a device that implements the RUNBIST instruction describes the duration of the test, the state of the output pins when the test is run, and the expected results.

The SCANSTA112 supports special execution sequences that simplify the operation of running BIST on a LSP scan chain. The sequence of instructions to run BIST testing on a parked SCANSTA112 port is as follows:

1. Pre-load the boundary register of the device under test if needed. This is sometimes necessary in BIST testing if the output pins of the device to be tested are driven by the values in the boundary register. This is usually noted in the BSDL file for the target device if it is required.
2. Issue the CNTRSEL instruction to the SCANSTA112 and initialize (load) the TCK counter to 00000000 Hex. Note that the TCK counter is initialized to 00000000 Hex when the SCANSTA112 TAP controller enters the *Test-Logic-Reset* state. This step may not be necessary if the TCK counter is already initialized to 00000000 Hex.
3. Issue the CNTRON instruction to the SCANSTA112 to enable the TCK counter. Note that the TCK counter will be 00000000 Hex at this time so it will have already reached its terminal count and will not produce additional clock pulses.
4. Shift the PARKRTI instruction into the SCANSTA112 instruction register and the RUNBIST instruction into the instruction register of the device under test. With the counter on, but at its terminal count and the LSP parked, the local TCK will be gated.
5. Issue the CNTRSEL instruction to the SCANSTA112.
6. Load the TCK counter. Shift the 32-bit value representing the number of TCK_n cycles needed to execute the BIST operation into the TCK counter register. The self test will begin on the rising edge of TCK_B following the *Update-Data-Register* TAP controller state.
7. Bit 7 of mode register 0 can be scanned to check the status of the TCK counter. This is accomplished by a MODESEL instruction followed by a *Shift-Data-Register* TAP controller state. The LSP currently executing its self test is parked, so it will not participate in this *Shift-Data-Register* operation. A logic 0 in bit 7 of mode register 0 means the counter has not reached terminal count. When the counter reaches its terminal count, bit 7 of mode register 0 will be set to logic 1. This means that the counter has reached terminal count and the BIST operation has completed.
8. Execute the CNTROFF instruction.
9. Unpark the LSP and scan out the result of the BIST operation. The register containing the results of the BIST operation is device-specific.

While a given LSP, or a set of LSPs, may be busy executing a built-in self test, other LSPs can perform other operations. It is a requirement of the IEEE 1149.1 standard that extra clock cycles beyond the number required for the RUNBIST command execution will be ignored. As long as the TCK counter register contains a value greater than the maximum number of cycles required by any device for its BIST to run, the extra cycles will be ignored by other devices in the scan chain.

It is possible to set up a built-in self test on one or more LSPs, park the LSPs so that the BIST will begin, perform boundary scan operations on devices on other LSPs, and then return to the parked LSPs after the BIST has completed to read out the results. Use of the TCK counter in the SCANSTA112 simplifies this procedure as described above.

10.4 SOFTRESET

The SOFTRESET instruction is provided to perform a Level 2 reset of all LSPs of selected SCANSTA112s. SOFTRESET forces all TMS_n signals high, placing the corresponding local TAP Controllers in the *Test-Logic-Reset* state within five TCK_B cycles. When the SOFTRESET instruction is issued, all LSPs are parked in the *Parked-TLR* state. To bring them out of this state, the SCANSTA112 must receive the UNPARK instruction and must subsequently be cycled through the *Run-Test/Idle* state.

The SOFTRESET instruction resets all the LSPs but does not reset the SCANSTA112. The SCANSTA112 remains in its selected state following the SOFTRESET instruction. The values previously written to the registers of the SCANSTA112 are not reset. The values set from the SCANSTA112's pins upon a reset are also not reset from the device pins following a SOFTRESET.

10.5 Port Synchronization

The topic of port synchronization has been discussed in conjunction with the discussion of parking LSPs. When a LSP is parked, it is placed in one of the four stable TAP Controller states: *Test-Logic-Reset*, *Run-Test/Idle*, *Pause-Data-Register*, or *Pause-Instruction-Register*. The SCANSTA112 is able to park a local scan chain by controlling its local Test Mode Select output ($TMS_{(0-6)}$) (see Figure 2). TMS_n is forced high for parking in the *Test-Logic-Reset* state, and forced low for parking in *Run-Test/Idle*, *Pause-Instruction-Register*, or *Pause-Data-Register* states. Taking a LSP out of its parked state and reinserting it into the active scan chain is accomplished by issuing the UNPARK instruction or, for a LSP parked in one of the pause states, the PARKPAUSE instruction. The LSPs do not become unparked until the SCANSTA112 TAP Controller is sequenced through a specified synchronization state. Synchronization occurs in the *Run-Test/Idle* state for LSPs parked in *Test-Logic-Reset* or *Run-Test/Idle*; and in the *Pause-Data-Register* or *Pause-Instruction-Register* state for ports parked in *Pause-Data-Register* or *Pause-Instruction-Register* respectively.

Figure 12 shows the waveforms for synchronization of a LSP that was parked in the *Test-Logic-Reset* state. Figure 13 shows the waveforms for synchronization of a LSP that was parked in the *Run-Test/Idle* state.

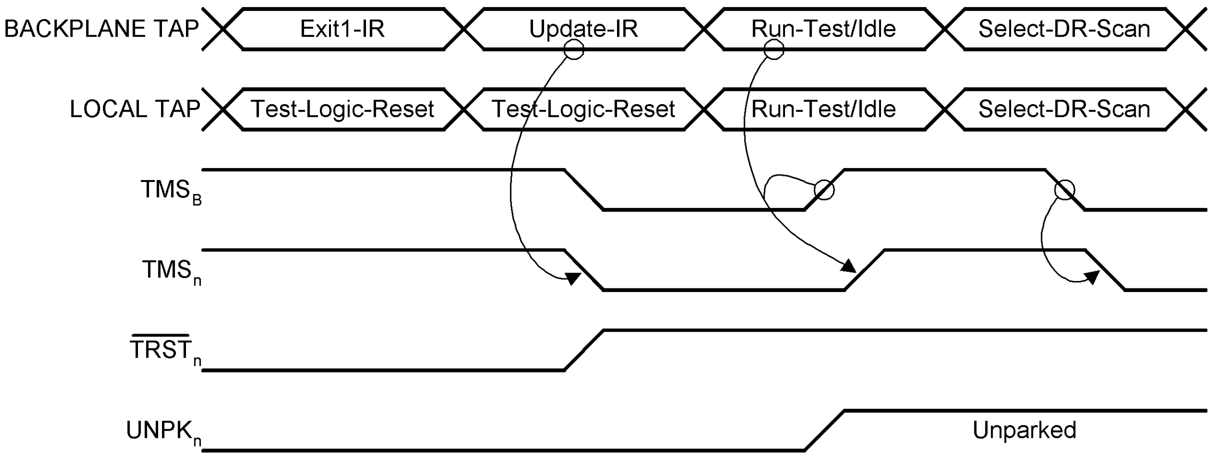

Figure 12. Local Scan Port Synchronization from Parked-TLR State

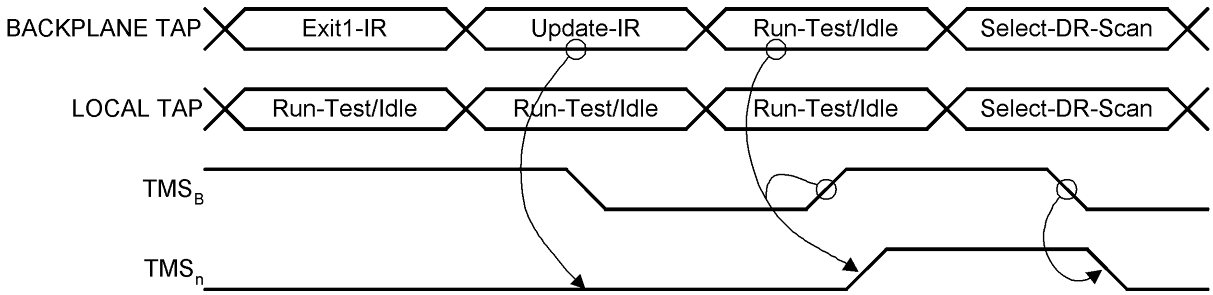

Figure 13. Local Scan Port Synchronization from Parked-RTI State

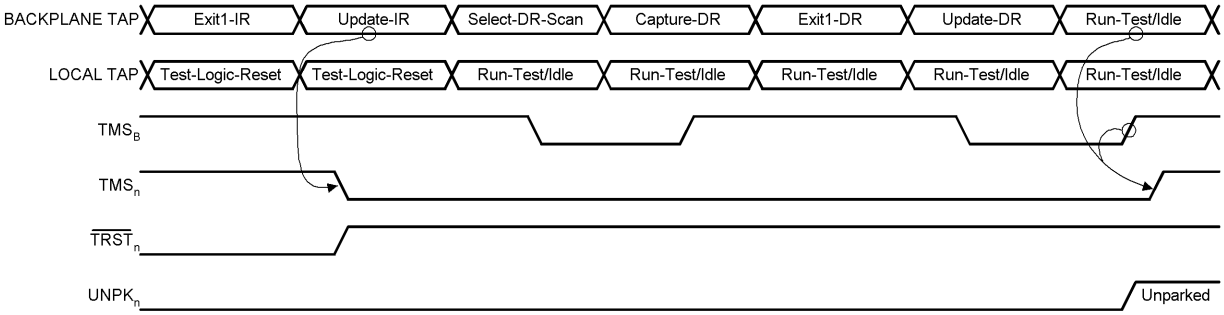

Figure 14. Local Scan Port Synchronization on Second Pass

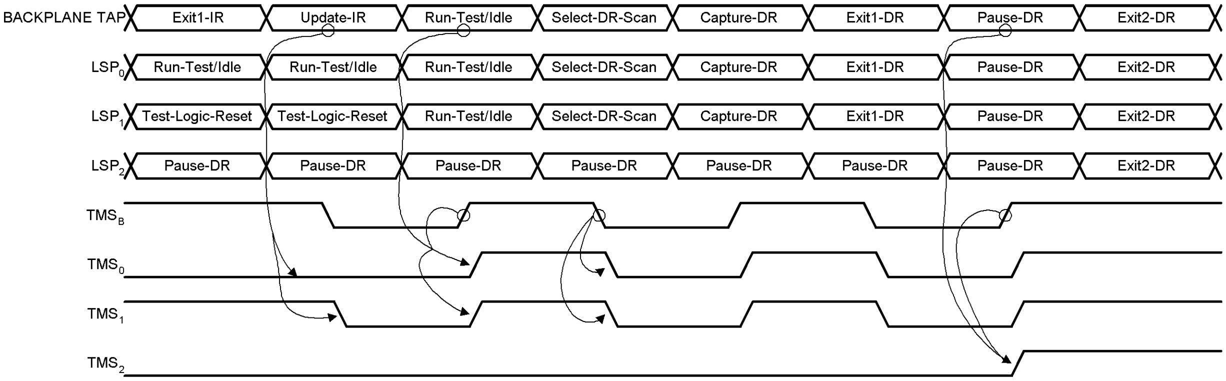

Figure 15. Synchronization of the Three Local Scan Ports

Once the UNPARK instruction is received in the instruction register, the local scan port controller forces TMS_n low on the falling edge of TCK_B. This moves the LSP TAP controllers to the synchronization state (*Run-Test/Idle*) where they remain until synchronization occurs. If the next state of the SCANSTA112 TAP Controller is *Run-Test/Idle*, TMS_n is connected to TMS_B and the LSP TAP Controllers are synchronized to the SCANSTA112 TAP controller as shown in Figure 12 and Figure 13.

If the next state after *Update-Instruction-Register* is *Select-Data-Register-Scan*, TMS_n remains low and synchronization does not occur until the SCANSTA112 TAP controller enters the *Run-Test/Idle* state, as shown in Figure 14. In short, a parked LSP stays parked until the SCANSTA112 TAP controller is sequenced through the state in which the LSP was parked.

Each local port has its own local scan port TAP controller. This is necessary because the LSP TAP controllers may not be in the same state at all times as the SCANSTA112 TAP controller or the backplane master TAP controller. The local scan port TAP controllers hold the local scan ports in a stable state when they are parked so that they can be re-synchronized with the backplane TAP controller when they are unparked.

The LSPs can be unparked in 128 different configurations. The fact that it is possible to chain multiple local scan ports presents not only the task of synchronizing the SCANSTA112 TAP Controller with the TAP Controllers of each individual local port, but also of synchronizing the individual local ports to one another.

When multiple local ports are selected for access, it is possible that two ports are parked in different states. This could occur if previous operations accessed the two ports separately and parked them in the two different states. The LSP configuration state machines and LSP TAP controllers handle this situation gracefully. Figure 15 shows the UNPARK instruction being used to access LSP_0, LSP_1, and LSP_2 in series (mode register 0 = XXX0X111 binary). LSP_0 and LSP_1 become active as the SCANSTA112 controller is sequenced through the *Run-Test/Idle* state. LSP_2 remains parked in the *Pause-Data-Register* state until the SCANSTA112 TAP Controller is sequenced through the *Pause-Data-Register* state. At that point, all the selected local ports are synchronized and are part of the active scan chain.

10.6 Power-Off High Impedance Inputs and Outputs

The SCANSTA112 backplane test port features power-off high impedance inputs and outputs.

The TDI_B, TMS_B, and $\overline{\text{TRST}}_B$ inputs have a 25KΩ pull-up resistor to V_{DD}. When the device is powered off (V_{DD} floating), these inputs appear to be a capacitive load to ground. When V_{DD} = 0V (*i.e.* not floating but tied to V_{SS}) these inputs appear to be capacitive with the 25KΩ pull up resistor in parallel with the capacitor to ground.

The TCK_B input has no pull-up resistor. When the device is powered off (V_{DD} floating), the input appears to be a capacitive load to ground. When V_{DD} = 0V (*i.e.* not floating but tied to V_{SS}) the input also appears to be a capacitive load to ground.

When the device is powered off (V_{DD} = 0V or floating), the TDO_B output appears to be a capacitive load.

Refer to the device IBIS model on the TI website for more details about the I/O characteristics.

10.7 $\overline{\text{TRST}}$

$\overline{\text{TRST}}_B$: Assertion of $\overline{\text{TRST}}_B$ will return the device back to its known power-up state. This reset can be masked so that it does not reset the SCANSTA112 by setting the IgnoreReset bit in the control register to 1.

$\overline{\text{TRST}}_n$:$\overline{\text{TRST}}_n$ is an output on the LSP side of the SCANSTA112. While the local scan port selection state machine is in the *Parked-TLR* state the $\overline{\text{TRST}}_n$ pin will be driven low. In all other parked states the $\overline{\text{TRST}}_n$ pin will be driven high. The TLR_TRST input pins of the SCANSTA112 can be used to change the value driven on the $\overline{\text{TRST}}_n$ pin when the LSP is in the *Parked-TLR* state.

10.8 TRI-STATE Indicator Outputs

TRIST for TDO_{B0}, TDO_{B1}, and $\text{TDO}_{(1\text{–}3)}$ are signals for enabling an external buffer circuit between the SCANSTA112 and the backplane/LSP. This would allow, for example, a CMOS-to-LVDS converter to drive an LVDS JTAG backplane test bus. These signals are always driving. A separate TRIST output is provided for the each of the two backplane TAP ports and for three of the local scan ports, LSP_1, LSP_2, and LSP_3. These lines are asserted to report a tri-state condition on the associated TDO pin.

10.9 Pass-Through Pins

Each of the backplane TAP ports, BP0 and BP1, and LSP_1, are associated with two input and two output pass-through pins each. Each pair of pass-through pins consists of an input (A_n) and an output (Y_n). The LSP pass-through output (Y_n) will drive the level being received by the backplane pass-through input (A_B). Conversely, the level on the LSP pass-through input (A_n) will be driven on the backplane pass-through output (Y_B).

The pass-through pins are available only when a single LSP, either LSP_0 or LSP_1 is selected. For each of these LSPs, these pins will be enabled when the local scan port selection state machine is not in the *Parked-TLR* state. When the output pins are not enabled they will be at TRI-STATE.

A simple way to understand this operation is to consider that the SCANSTA112 really has eight JTAG TAPs. Let’s number them as follows:

- BP0 – backplane master or local scan slave
- BP1 – backplane master or local scan slave
- 01 – local scan slave
- 02 – local scan slave
- 03 – local scan slave
- 04 – local scan slave
- 05 – local scan slave
- 06 – local scan slave

The first three ports each have two pass-through outputs and two pass-through inputs associated with them. These pins are designated in the SCANSTA112 datasheet as $A0_{B0}$, $A1_{B0}$, $A0_{B1}$, $A1_{B1}$, $Y0_{B0}$, $Y1_{B0}$, $Y0_{B1}$, $Y1_{B1}$, $A0_{01}$, $A1_{01}$, $Y0_{01}$, and $Y1_{01}$. In this nomenclature, “A” always means an input pin and “Y” always means an output pin. The second character is either “0” or “1” and designates which of the two pass-through input/output pin pairs is referred to. The last two characters are either “B0”, meaning backplane port 0 (BP0), “B1”, meaning backplane port 1 (BP1), or “01”, meaning LSP_1 (01), the third item on the list above. No matter what other configuration settings are made in the SCANSTA112, this is the way these pins are mapped. This is the key point to understanding how the pass-through inputs and outputs work.

With $MPsel_{B1/\overline{B0}}$ tied low, port BP0 is the backplane master port and port BP1 is local scan port 0 (LSP_0). That means pins $A0_{B1}$, $A1_{B1}$, $Y0_{B1}$, and $Y1_{B1}$ are associated with local scan port 0. When local scan port 0 is selected (mode register 0 = 01 Hex), then the inputs and outputs are mapped as shown in Table 14.

Similarly, when local scan port 1 is selected (mode register 0 = 02 Hex), then the inputs and outputs are mapped as shown in Table 15.

Table 14. Mapping of Pass-Through Pins with $MPsel_{B1/\overline{B0}}$ at Logic 0 and LSP_0 Selected

Input	Passes Through To	Output	Comment
$A0_{B0}$	→	$Y0_{B1}$	$A0_{B0}$ is the backplane input, $Y0_{B1}$ is LSP_0 output
$A1_{B0}$	→	$Y1_{B1}$	$A1_{B0}$ is the backplane input, $Y1_{B1}$ is LSP_0 output
$A0_{B1}$	→	$Y0_{B0}$	$A0_{B1}$ is LSP_0 input, $Y0_{B0}$ is the backplane output
$A1_{B1}$	→	$Y1_{B0}$	$A1_{B1}$ is LSP_0 input, $Y1_{B0}$ is the backplane output
$A0_{01}$	→	Nowhere	$A0_{01}$ is LSP_1 input which is not selected
$A1_{01}$	→	Nowhere	$A1_{01}$ is LSP_1 input which is not selected
High-Z	→	$Y0_{01}$	$Y0_{01}$ is LSP_1 output which is not selected
High-Z	→	$Y1_{01}$	$Y1_{01}$ is LSP_1 output which is not selected

Table 15. Mapping of Pass-Through Pins with $MPsel_{B1/\overline{B0}}$ at Logic 0 and LSP_1 Selected

Input	Passes Through To	Output	Comment
$A0_{B0}$	→	$Y0_{01}$	$A0_{B0}$ is the backplane input, $Y0_{01}$ is LSP_1 output
$A1_{B0}$	→	$Y1_{01}$	$A1_{B0}$ is the backplane input, $Y1_{01}$ is LSP_1 output
$A0_{01}$	→	$Y0_{B0}$	$A0_{01}$ is LSP_1 input, $Y0_{B0}$ is the backplane output
$A1_{01}$	→	$Y1_{B0}$	$A1_{01}$ is LSP_1 input, $Y1_{B0}$ is the backplane output
$A0_{B1}$	→	Nowhere	$A0_{B1}$ is LSP_0 input which is not selected
$A1_{B1}$	→	Nowhere	$A1_{B1}$ is LSP_0 input which is not selected
High-Z	→	$Y0_{B1}$	$Y0_{B1}$ is LSP_0 output which is not selected
High-Z	→	$Y1_{B1}$	$Y1_{B1}$ is LSP_0 output which is not selected

The operation of the pass-through pins can be described using an example as follows. With $MPsel_{B1/\overline{B0}}$ is tied low, meaning BP0 is the master backplane port, and with LSP_1 selected (not LSP_0), backplane input $A0_{B0}$ goes to local scan port output $Y0_{01}$; backplane input $A1_{B0}$ goes to local scan port output $Y1_{01}$; local scan port input $A0_{01}$ goes to backplane output $Y0_{B0}$; and local scan port input $A1_{01}$ goes to backplane output $Y1_{B0}$.

When more than one local scan port is selected, all the outputs go to a high impedance state and none of the inputs goes anywhere.

If $MPsel_{B1/\overline{B0}}$ is tied high instead of low, the roles of the backplane inputs and outputs are switched with those of local scan port 0.

10.10 LSP TCK Gating

While the LSP_n local scan port selection state machine is in the *Parked-TLR* state, the four LSP_n signals will be controlled as shown in Table 16. Upon entry into the *Parked-TLR* state (when the SCANSTA112 is powered up or reset or when the PARKTLR or GOTOWAIT instructions are received) a counter in the LSP_n state machine allows 512 TCK_B clock pulses to occur on TCK_n before gating TCK_n low. Once gated, TCK_n will drive a logic 0.

The purpose of this operation is to reset all the SCANSTA112s in a hierarchical scan chain. Letting 512 TCK_B pulses pass through to TCK_n allows a five high TMS reset to occur on over 100 levels of hierarchy before the SCANSTA112 gates TCK_n, which saves power versus a free-running clock system.

Table 16. Gated LSP Drive States

LSP Connection	Drive State
TDO_n	Driven HIGH – drive capability 12 mA
TMS_n	Driven HIGH – drive capability 24 mA
TDI_n	Pull-up resistor to provide a weak HIGH
TCK_n	TCK_B for 512 pulses, then gated LOW

The SCANSTA112 does not require that any clock pulses be received on TCK_B while in the *Parked-TLR* state. In order to enable the TCK gating behavior shown in Table 16, however, clock pulses on the backplane TCK_B input should not be gated immediately after placing the SCANSTA112 in the *Parked-TLR* state.

Setting Bit 3 of mode register 0 to 1 gates TCK_n when in the *Parked-RTI*, *Parked-Pause-DR* and *Parked-Pause-IR* states. Default is free-running (bit 3 = 0). The value stored in bit 3 of mode register 0 does not affect the propagation of 512 TCK_B clock pulses before gating TCK_n in the *Parked-TLR* state. (See section on mode register 0).

10.11 GPIO Connections

General Purpose I/O (GPIO) pins are registered inputs and outputs. The GPIO operation of the SCANSTA112 is described below.

In the shared GPIO mode of operation, the IEEE 1149.1 LSP pins TDI_n, TDO_n and TMS_n pins become GPIO pins. TMS_n and TDO_n will be outputs, TDI_n will be an input in the GPIO mode.

The sequence of operations to use shared GPIOs on a LSP are as follows (the example uses LSP_0). GPIO operation is only available when the SCANSTA112 is in ScanBridge mode.

1. Scan the SCANSTA112 address into the instruction register (address a SCANSTA112 in direct address mode).
2. Scan the MODESEL2 instruction into the instruction register to select mode register 2 (shared GPIO configuration register) as the data register.
3. Scan 00000001 into mode register 2 to enable shared GPIO operation on LSP_0. The GPIO will be enabled when the TAP enters the *Run-Test/Idle* state at the end of this shift operation (TDO_0 and TMS_0 will be forced to logic "0" as defined by the default value in the Shared GPIO $Register_0$).
4. Scan the SGPIO0 instruction into the instruction register to select the shared GPIO register 0 as the data register.
5. Scan 00000011 into the shared GPIO register 0 to set TDO_0 and TMS_0 to a logic "1" (when the TAP enters *Update-Data-Register*). During this operation, when the TAP enters *Capture-Data-Register*, the present value on the TDI_0 pin and the values of TDO_0 and TMS_0 (as set by shared GPIO register 0) will be captured into bits 2, 1 and 0 of the shift register and will be scanned out 00000X00 (X = value

present on TDI_0 when TAP enters *Capture-Data-Register*). Note that the output bits are ones-complemented.

6. Step 5 can be repeated to generate waveforms on TDO_0 and TMS_0. If step 5 was repeated with 00000000 as data, TDO_0 and TMS_0 would be set to a logic 0 (when TAP state = *Update-Data-Register*) and 00000X11 would be scanned out (X = value present on TDI_0 when TAP enters *Capture-Data-Register*).
7. Scan the GOTOWAIT or SOFTRESET instruction, or generate a $\overline{TRST_B}$ reset to disable the GPIO.

10.12 Address Mask

This feature masks S_5-S_0 and allows the SCANSTA112 to respond to an address on pins S_7-S_6 only. For example if ADDMASK is deasserted, a SCANSTA112's address could be F3 Hex or 11110011 binary. With ADDMASK asserted, the same SCANSTA112 would respond to any address in the range of C0 Hex to FF Hex or 11000000 binary to 11111111 binary. This feature is used when a local boundary scan controller utilizes one address space and an alternate boundary scan controller such as a manufacturing test system uses a second address space. This feature enables identical cards with local controllers to be selected using a two-bit address while responding to unique 8-bit addresses in the operational system. Once the system boundary scan controller is connected and ADDMASK is deasserted each SCANSTA112 then has a unique address.

The broadcast, multi-cast, and interrogate addresses are reserved and are still recognized when the address mask line is asserted.

10.13 Address Interrogation

The SCANSTA112 selection state machine has four states that it can go to from the *Wait-For-Address* state: *Unselected*, *Selected-Single-SCANSTA112*, *Selected- Multiple-SCANSTA112*, and *Address-interrogation* (see Figure 16).

After a reset (or GOTOWAIT command) has been issued, the SCANSTA112 TAP is sequenced to the *Capture-Instruction-Register* state where XXXXXX01 is loaded into the shift register. At this point the SCANSTA112 is looking for an address match in its instruction register. In order to interrogate the addresses of all the SCANSTA112s in the system the address interrogation “address” is used.

Upon entering the *Shift-Instruction-Register* state, the boundary scan controller scans the address interrogation value (3A hex) into the instruction registers of all the SCANSTA112s in the system. This value is loaded into the address registers of all the SCANSTA112s as the TAP is sequenced into the *Update-Instruction-Register* state. When the TAP controller again passes through the *Capture-Instruction-Register* state, the shift register of each SCANSTA112 is loaded with the ones-complement of bits 5–0 of the slot address of that SCANSTA112. In the *Shift-Instruction-Register* state the ones-complement of the slot address of the SCANSTA112 is shifted out on the TDO line. While this shift operation is performed the TDO_B driver is set to drive strong 0s and weak 1s. The value presented on TDO_B will be a wired-and of bits 5-0 of the addresses of all of the SCANSTA112s on the bus.

As the address value of each SCANSTA112 is shifted out, each SCANSTA112 will monitor its TDO_B to see if it is receiving the same value it is driving. If the SCANSTA112 shifts all the bits of its ones-complement address and never gets a compare error it will tri-state its TDO_B and go to the *Wait-For-Reset* state. Alternately, if the SCANSTA112 sees a compare error while it is shifting its ones-complement address it will stop shifting its address and tri-state its TDO_B until the next shift instruction register operation. Then, when the TAP again enters the *Shift-Instruction-Register* state it will again try to present its address (if the previous instruction was 3A hex) while monitoring TDO_B.

Shifting 3A hex into the instruction registers of the SCANSTA112s will continue until all SCANSTA112s have presented their addresses. At this time all the SCANSTA112s in the system will be waiting to be reset, and if a 3A is shifted into the SCANSTA112 instruction registers the address read by the boundary scan controller will be all weak 1s because the TDO_B lines of all the SCANSTA112s will be at TRI-STATE. Reading all ones will signal the boundary scan controller that address interrogation is complete.

Since a reading of all ones signifies the end of *Address-Interrogation*, no device can have an address of all zeros (the ones-complement of all ones). Because only bits 5–0 of the SCANSTA112 slot address are scanned out during address interrogation, each SCANSTA112 in the system should be assigned a slot address in the range from 01 Hex to 39 Hex.

If at any time during the address interrogation mode, any other instruction besides 3A hex is shifted into the instruction register, then the SCANSTA112s will exit the interrogation mode and will go to the *Wait-For-Address* state.

This address interrogation scheme presumes that TDO_B is capable of driving a weak 1 and that an SCANSTA112 driving a 0 will overdrive a SCANSTA112 driving a weak 1. For address interrogation to work, the TDO_B outputs of all the SCANSTA112s in the system should have weak pull-up resistors (25KΩ, for example) to V_{DD}.

The following is an example of the *Address-Interrogation* function. Assume there are three SCANSTA112s (U1, U2 and U3) on an 1149.1 backplane. The three SCANSTA112s have slot addresses 00010100, 00100000 and 00000001 respectively.

1. The SCANSTA112s are reset and the interrogation address/opcode (3A hex) is shifted into the instruction registers.
2. At the end of the instruction shift (*Update-Instruction-Register*) the SCANSTA112 address registers are loaded with 3A hex.
3. The TAP is sequenced to *Capture-Instruction-Register* and the instruction registers latch the ones-complement slot addresses (U1=00101011, U2=00011111 and U3=00111110).
4. The TAP is sequenced to *Shift-Instruction-Register* and the LSB of the interrogation address is again presented on the TDI_B line. Concurrently, the LSBs of the ones-complement slot addresses are presented by all the SCANSTA112s on the TDO_B line.
5. The weak 1 being driven on U1 and U2 is overdriven by the 0 from U3. U1 and U2 enter the *Wait-For-Next-Interrogation* state.
6. The shift operation continues and U3 finishes shifting its ones-complement address (00111110) out on TDO_B. U3 enters the *Wait-For-Reset* state when the TAP enters *Update-Instruction-Register*.
7. The TAP is again sequenced to *Capture-Instruction-Register* and U1 and U2 shift registers latch the ones-complement addresses (U1=00101011, U2=00011111).
8. The TAP is sequenced to *Shift-Instruction-Register* and the LSB of the interrogation address is presented on the TDI_B line. Concurrently, the LSBs of the ones-complement slot addresses are presented by U1 and U2 on the TDO_B line.
9. Since both U1 and U2 are driving a weak 1 the shift continues.
10. Again U1 and U2 drive weak 1 and the shift continues.
11. U2's weak 1 is overdriven by U1's 0 and U2 enters the *Wait-For-Next-Interrogation* state.
12. The shift operation continues and U1 finishes shifting its ones-complement address (00101011) out on TDO_B. U1 enters the*Wait-For-Reset* state.
13. The instruction shift operation is repeated and U2 shifts its ones-complement address (00011111) out on TDO_B. U2 enters the *Wait-For-Reset* state.
14. The instruction shift operation is repeated. At this point, all devices have been interrogated and are waiting for a reset. The boundary scan controller will receive all ones. If there is not a SCANSTA112 with address 00000000 then the address interrogation operation is complete and all the SCANSTA112s in the system have reported their addresses to the boundary scan controller.

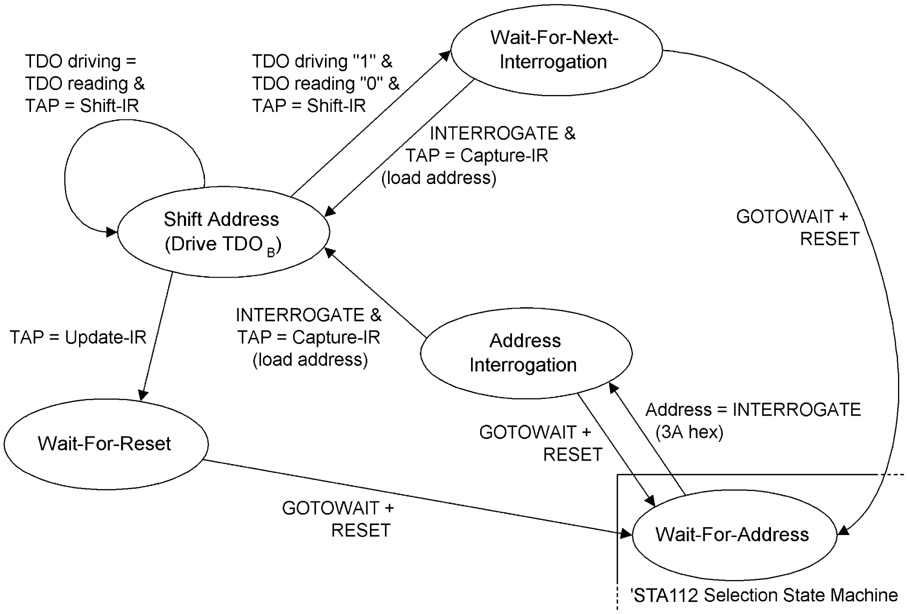
Wait-For-Next-Interrogation
TDO driving = TDO reading & TAP = Shift-IR
TDO driving "1" & TDO reading "0" & TAP = Shift-IR
INTERROGATE & TAP = Capture-IR (load address)
Shift Address (Drive TDO B)
GOTOWAIT + RESET
TAP = Update-IR
INTERROGATE & TAP = Capture-IR (load address)
Address Interrogation
Address = INTERROGATE (3A hex)
GOTOWAIT + RESET
Wait-For-Reset
GOTOWAIT + RESET
Wait-For-Address
'STA112 Selection State Machine

Figure 16. Address Interrogation State Machine

11 Terms and Abbreviations

The following table summarizes many of the terms and abbreviations used in this document.

Table 17. Glossary

LFSR	Linear Feedback Shift Register. When enabled, will generate a 16-bit signature of sampled serial test data.
LSP	Local Scan Port. A five signal port that drives a local (*i.e.* non-backplane) scan chain. *(e.g.*, TCK_0, TMS_0, TDO_0, TDI_0, $\overline{TRST_0}$).
Local	Local is used to describe IEEE Std. 1149.1 compliant scan chains and the SCANSTA112 Test Access Port that drives them. The term local was adopted from the system test architecture that the SCANSTA112 will most commonly be used in: namely, a system test backplane with a SCANSTA112 on each card driving up to 7 local scan chains per card. (Each card can contain multiple SCANSTA112s, with 7 local scan ports per SCANSTA112.)
Park/Unpark/Unparked	Parked, unpark, and unparked, are used to describe the state of the LSP controller and the state of the local TAP controllers (a local TAP controllers is the TAP controller of the scan components that make up a local scan chain). Park is also used to describe the action of parking a LSP (transitioning into one of the parked LSP controller states). It is important to understand that when a LSP controller is in one of the parked states, TMS_n is held constant, thereby holding or parking the local TAP controllers in a given state.
TAP	Test Access Port as defined by IEEE Std. 1149.1.
Selected/Unselected	Selected and Unselected refers to the state of the SCANSTA112 Selection Controller. A selected SCANSTA112 has been properly addressed and is ready to receive Level 2 protocol instructions. Unselected SCANSTA112s monitor the system test backplane, but do not accept Level 2 protocol instructions (except for the GOTOWAIT instruction). The data registers and LSPs of unselected SCANSTA112s are not accessible from the boundary scan controller.
Active Scan Chain	The Active Scan Chain refers to the scan chain configuration as seen by the boundary scan controller at a given moment. When a SCANSTA112 is selected with all of its LSPs parked, the active scan chain is the current scan register only. When a LSP is unparked, the active scan chain becomes: TDI_B → the current SCANSTA112 register → the local scan chain registers → a PAD bit → TDO_B.
Level 1 Protocol	Level 1 is the protocol used to address a SCANSTA112.
Level 2 Protocol	Level 2 is the protocol that is used once a SCANSTA112 is selected. Level 2 protocol is IEEE Std. 1149.1 compliant when an individual SCANSTA112 is selected.
PAD	A one-bit register that is placed at the end of each local scan port's scan chain. The PAD bit eliminates the propagation delay that would be added by the SCANSTA112 LSP_n logic between TDI_n and $TDO_{(n+1)}$ or TDO_B by buffering and synchronizing the LSP TDI inputs to the falling edge of TCK_B, thus allowing data to be scanned at higher frequencies without violating setup and hold times.

Table 17. Glossary (continued)

LSB	Least-Significant Bit, the right-most position in a register (bit 0).
MSB	Most-Significant Bit, the left-most position in a register.
Transparent	A mode of operation which eliminates any SCANSTA112 registers or pad bits from the scan chain.
Stitcher	A mode of operation where LSPs to include in the scan chain may be defined by external pins, or by the contents of a register.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265