National Semiconductor

November 2006

SCANSTA101
Low Voltage IEEE 1149.1 STA Master

General Description

The SCANSTA101 is designed to function as a test master for a IEEE 1149.1 test system. The minimal requirements to create a tester are a microcomputer (uP, RAM/ROM, clock, etc.), SCANEASE r2.0 software, and a STA101.

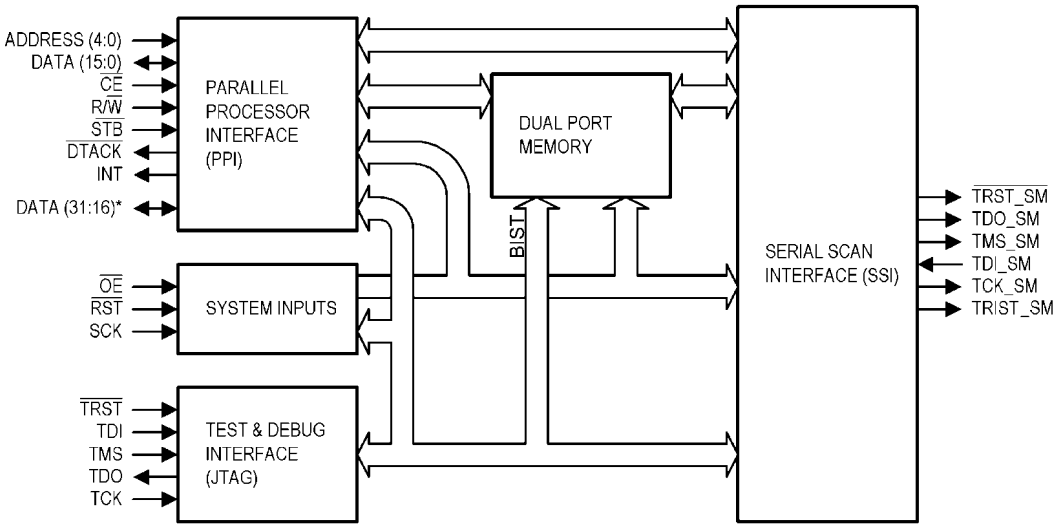
The SCANSTA101 is an enhanced version of, and replacement for, the SCANPSC100. The additional features of the STA101 further allow it to offload some of the processor overhead while remaining flexible. The device architecture supports IEEE 1149.1, BIST, and IEEE 1532. The flexibility will allow it to adapt to any changes that may occur in 1532 and support yet unknown variants.

The SCANSTA101 is useful in improving vector throughput when applying serial vectors to system test circuitry and reduces the software overhead that is associated with applying serial patterns with a parallel processor. The SCANSTA101 features a generic Parallel Processor Interface (PPI) which operates by serializing data from the parallel bus for shifting through the chain of 1149.1 compliant components (i.e., scan chain). Writes can be controlled either by wait states or the $\overline{\text{DTACK}}$ line. Handshaking is accomplished with either polling or interrupts.

Features

- Compatible with IEEE Std. 1149.1 (JTAG) Test Access Port and Boundary Scan Architecture
- Supported by National's SCAN Ease (Embedded Application Software Enabler) Software Rev 2.0
- Uses generic, asynchronous processor interface; compatible with a wide range of processors and PCLK frequencies
- 16-bit Data Interface (IP scalable to 32-bit)
- 2Kx32 bit dual-port memory addressing for access by the PPI or the 1149.1 master
- Load-on-the-fly (LotF) and Preload operating modes supported
- On-Board Sequencer allows multi-vector operations such as those required to load data into an FPGA
- On-Board Compares support TDI validation against preloaded expected data
- 32-bit Linear Feedback Shift Register (LFSR) at the Test Data In (TDI) port
- State, Shift, and BIST macros allow predetermined TMS sequences to be utilized
- Operates at 3.3v supply voltages w/ 5V tolerant I/O
- Outputs support Power-Down TRI-STATE mode.

SCANSTA101 Architecture



* Note: IP data bus width can be configured for 16 or 32 bit applications. Silicon data bus width is fixed at 16 bits.

10121502

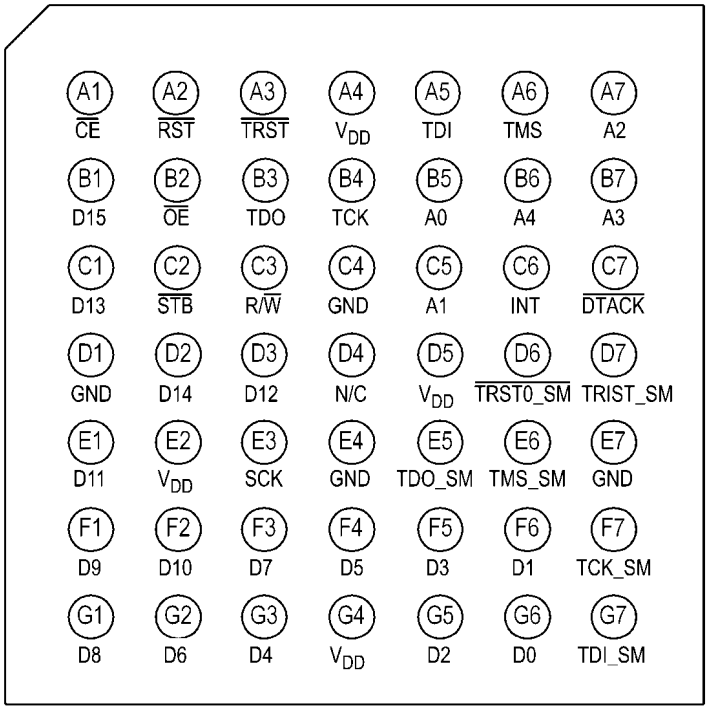
FIGURE 1.

Figure 1 shows a high level view of the SCANSTA101 Scan Master and its interface groups. *Table 1* provides a brief description of each of these interface groups. *Table 2* provides a brief description of the external interfaces. The device is composed of three interfaces around a dual-port memory. These interfaces consist of the Parallel Processor Interface (PPI), Serial Scan Interface (SSI), and Test and Debug Interface. The System Input block is included only to designate inputs that have global use across the device. The Test and Debug interface supports BIST, boundary scan, and internal scan for this device.

TABLE 1. Interface Descriptions

Interface	Description
Parallel Processor Interface	Used for configuration, ScanMaster scan chain loads and reads, programmable device file loads and reads, and status monitoring.
Serial Scan Interface	Performs parallel to serial conversion, sequences and formats the outgoing serial stream to conform to 1149.1 protocol.
Test and Debug Interface	Interfaces used for manufacturing tests, this includes a JTAG interface and a scan interface. The three scan interface pins are shared with three of the data pins.
System Inputs	Interface inputs for system control, i.e. clock, reset and output tristate control.

Connection Diagram



10121540

BGA Package Pinout
(Top View)

TABLE 2. Pin Descriptions

Pin Name	No. Pins	I/O	Description
VCC	4	N/A	Power
GND	4	N/A	Ground
D(15:0)	16	I/O	Bidirectional Data Bus. Signals are bonded out for the packaged device. D15 and D14 are shared pins with SCAN_IN, and SCAN_OUT respectively.
D(31:16) (Note 1)	16	I/O	Bidirectional Data Bus. These signals are not available in the packaged device.
A(4:0)	5	I	Address Bus
SCK	1	I	The system clock that drives all internal timing. TCK_SM is a gated, divided and buffered version of SCK.
INT	1	O	Interrupt Output
$\overline{OE}$	1	I	Output enable that tristates all 1149.1 "_SM" outputs when high.
$\overline{DTACK}$	1	O	$\overline{DTACK}$ is used to synchronize asynchronous transfers between the host and the STA101. When $\overline{CE}$ is high, $\overline{DTACK}$ is tristated. When $\overline{CE}$ is low, $\overline{DTACK}$ is enabled. $\overline{DTACK}$ goes low when data has been registered and then goes tri-state when the cycle has completed.
R/$\overline{W}$	1	I	R/$\overline{W}$ defines a PPI cycle. Read when high, write when low.
$\overline{STB}$	1	I	Strobe is used for timing all PPI transfers. D(15:0), or D(31:0) in 32-bit mode, are tristated when $\overline{STB}$ is high. Data valid setup is with respect to the falling edge of $\overline{STB}$ and data valid hold is with respect to rising edge of $\overline{STB}$.
$\overline{CE}$	1	I	Chip Enable, when low, enables the PPI for data transfers. $\overline{CE}$ can remain low during back-to-back accesses. D(15:0), or D(31:0) in 32-bit mode, and $\overline{DTACK}$ are tristated when $\overline{CE}$ is high.
$\overline{RST}$	1	I	Asynchronous reset, when low, initializes the STA101.
TDO	1	O	Test Data Out is the serial scan output from the STA101. TDO is enabled when $\overline{OE}$ is low.
TDI	1	I	Test Data In is the serial scan input to the STA101.
TMS	1	I	Test Mode Select. The Test Mode Select pin is a serial input used to accept control logic to the Test & debug interface.
TCK	1	I	Test Clock Input for 1149.1
$\overline{TRST}$	1	I	Test Reset. This pin should be tied to ground by a 1K resistor to hold the Test and Debug Interface in the Test-Logic-Reset state during device power-up. This avoids invalid states when ramping supply voltages.
TDI_SM	1	I	Scan Master Test Data Input in the Serial Scan Interface
TDO_SM	1	O	Scan Master Test Data Output in the Serial Scan Interface
TMS_SM	1	O	Scan Master Test Mode Select in the Serial Scan Interface
TCK_SM	1	O	Scan Master Test Clock in the Serial Scan Interface
$\overline{TRST0\_SM}$	1	O	Scan Master Test Reset output in the Serial Scan Interface
$\overline{TRST1\_SM}$ (Note 1)	1	O	Redundent ScanMaster $\overline{TRST}$. This signal is not available for the packaged device.
TRIST_SM	1	O	The TRI-STATE notification pin exerts a high signal when TDO_SM is TRI-STATED

Note 1: D(31:16) in the Parallel Processor Interface and TRST1_SM in the Serial Scan Interface are not bonded out for the packaged part. These are used in the 32-bit Macro Mode only.

Absolute Maximum Ratings (Note 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage (V_{CC})	−0.5V to +4.0V
DC Input Diode Current (I_{IK})	
V_I = −0.5V	−20 mA
DC Input Voltage (V_I)	−0.5V to +4.0V
DC Output Diode Current (I_{OK})	
V_O = −0.5V	−20 mA
DC Output Voltage (V_O)	−0.5V to +4.0V
DC Output Source/Sink Current (I_O)	±50 mA
DC V_{CC} or Ground Current per Output Pin	±50 mA
DC Latchup Source or Sink Current	±300 mA
Junction Temperature	
Plastic	+150°C
Storage Temperature	−65°C to +150°C
Lead Temperature (Solder, 4sec)	
49L BGA	220°C
Max Pkg Power Capacity @ 25°C	
49L BGA	1.47W
Thermal Resistance (θ_{JA})	
49L BGA	85°C/W
Package Derating	11.8mW/°C above +25°C
ESD Last Passing Voltage (Min)	2000V

Recommended Operating Conditions

Supply Voltage (V_{CC})	3.0V to 3.6V
Input Voltage (V_I)	0V to V_{CC}
Output Voltage (V_O)	0V to V_{CC}
Operating Temperature (T_A)	−40°C to +85°C

Note 2: Absolute maximum ratings are those values beyond which damage to the device may occur. The databook specifications should be met, without exception, to ensure that the system design is reliable over its power supply, temperature, and output/input loading variables. National does not recommend operation of SCAN STA products outside of recommended operation conditions.

DC Electrical Characteristics

Over recommended operating supply voltage and temperature ranges unless otherwise specified.

Symbol	Parameter	Conditions	Min	Max	Units
V_{IH}	Minimum High Input Voltage	V_{OUT} = 0.1V or V_{CC} − 0.1V	2.1		V
V_{IL}	Maximum Low Input Voltage	V_{OUT} = 0.1V or V_{CC} − 0.1V		0.8	V
V_{OH}	Minimum High Output Voltage	I_{OUT} = −100 µA, V_{IN} = V_{IL} or V_{IH}	V_{CC}-0.2V		V
	Minimum High Output Voltage, TDO_SM, TMS_SM, TCK_SM, $\overline{\text{TRST0\_SM}}$ outputs only	I_{OH} = −24 mA, V_{IN} = V_{IL} or V_{IH}	2.2		V
	Minimum High Output Voltage, All other outputs including 1149.1	I_{OH} = −12 mA, V_{IN} = V_{IL} or V_{IH}	2.4		V
V_{OL}	Maximum Low Output Voltage	I_{OUT} = +100 µA, V_{IN} = V_{IL} or V_{IH}		0.2	V
	Maximum Low Output Voltage, TDO_SM, TMS_SM, TCK_SM, $\overline{\text{TRST0\_SM}}$ outputs only	I_{OL} = 24 mA, V_{IN} = V_{IL} or V_{IH}		0.5	V
	Maximum Low Output Voltage, all other outputs including 1149.1	I_{OL} = 12mA, V_{IN} = V_{IL} or V_{IH}		0.4	V
I_{IN}	Maximum Input Leakage Current, All pins except TDI, TMS, $\overline{\text{TRST}}$, TDI_SM	V_{IN} = V_{CC} for TDI, $\overline{\text{OE}}$, V_{IN} = V_{CC}, GND for All Others		±5.0	µA
I_{ILR}	Maximum Input Leakage Current, TDI, TMS, $\overline{\text{TRST}}$, TDI_SM	V_{IN} = GND	-45	-200	µA
I_{IH}	Maximum Input Leakage Current, TDI, TMS, $\overline{\text{TRST}}$, TDI_SM	V_{IN} = V_{CC}		5.0	µA
I_{OZ}	Maximum TRI-STATE Leakage Current	V_{IN} = V_{CC}, GND, V_{IN} ($\overline{\text{OE}}$, R/$\overline{\text{W}}$, $\overline{\text{CE}}$, $\overline{\text{STB}}$) = VIL, VIH		±5.0	µA
I_{OFF}	Power Off Leakage Current All pins except TDI, TMS, $\overline{\text{TRST}}$, and TDI_SM	V_{CC} = 0.0V		5.0	µA
I_{CC}	Maximum Quiescent Supply Current			250	µA
I_{CCmax}	Maximum Supply Current	All inputs low		1.2	mA
I_{CCT}	Maximum I_{CC}/Input	V_{IN} = V_{CC} − 0.6V		250	µA

AC Electrical Characteristics/Operating Requirements

Over recommended operating supply voltage and temperature ranges unless otherwise specified. C_L = 50 pF, R_L = 500Ω unless otherwise specified.

Symbol	Parameter	Conditions	# of SCK (Notes 3, 4)	Min	Max	Units
PARALLEL PROCESSOR INTERFACE (PPI)						
t_{S1}	Set Up Time $\overline{CE}$, R/$\overline{W}$, Addr, Data to $\overline{STB}$	*Figures 11, 12*		0		ns
t_{H1}	Hold Time $\overline{CE}$, R/$\overline{W}$, Addr, Data to $\overline{DTACK}$	*Figures 11, 12*		0		ns
t_{D1}	Propagation Delay $\overline{STB}$ low to $\overline{DTACK}$ low, Register Write	*Figure 11*	2 or 3		11.5	ns
t_{D1}	Propagation Delay $\overline{STB}$ low to $\overline{DTACK}$ low, Register Read	*Figure 12*	4 or 5		11.5	ns
t_{D1}	Propagation Delay $\overline{STB}$ low to $\overline{DTACK}$ low, Memory Write: 16-bit first access	*Figure 11*	3 or 4		11.5	ns
t_{D1}	Propagation Delay $\overline{STB}$ low to $\overline{DTACK}$ low, Memory Write: 16-bit second access	*Figure 11*	7 or 8		11.5	ns
t_{D1}	Propagation Delay $\overline{STB}$ low to $\overline{DTACK}$ low, Memory Read: 16-bit first access	*Figure 12*	9 or 10		11.5	ns
t_{D1}	Propagation Delay $\overline{STB}$ low to $\overline{DTACK}$ low, Memory Read: 16-bit second access	*Figure 12*	3 or 4		11.5	ns
t_{D2}	Propagation Delay $\overline{STB}$ high to $\overline{DTACK}$ TRISTATE, Register Write	*Figure 11*	1 or 2		10.0	ns
t_{D2}	Propagation Delay $\overline{STB}$ high to $\overline{DTACK}$ TRISTATE, Register Read	*Figure 12*	1 or 2		10.0	ns
t_{D2}	Propagation Delay $\overline{STB}$ high to $\overline{DTACK}$ TRISTATE, Memory Write: 16-bit first access	*Figure 11*	1 or 2		10.0	ns
t_{D2}	Propagation Delay $\overline{STB}$ high to $\overline{DTACK}$ TRISTATE, Memory Write: 16-bit second access	*Figure 11*	1 or 2		10.0	ns
t_{D2}	Propagation Delay $\overline{STB}$ high to $\overline{DTACK}$ TRISTATE, Memory Read: 16-bit first access	*Figure 12*	1 or 2		10.0	ns
t_{D2}	Propagation Delay $\overline{STB}$ high to $\overline{DTACK}$ TRISTATE, Memory Read: 16-bit second access	*Figure 12*	1 or 2		10.0	ns
t_{D3}	Propagation Delay Output data valid to $\overline{DTACK}$ low, all read cycles	*Figure 12*	1			ns
t_{pHL1}	Propagation Delay $\overline{STB}$ low to INT low, register write (clears Interrupt)	*Figure 11*	5 or 6		10.5	ns
t_W	Clock Pulse Width, SCK, H or L			3.0		ns
f_{MAX}	Clock Frequency, SCK				66	MHz
$t_{RELEASE}$	Release Time, $\overline{RST}$ to $\overline{STB}$		2			ns

Note 3: Due to uncertainty in the relationship of the $\overline{STB}$ placement to the system clock, SCK, the $\overline{STB}$ may be detected during the current or the next SCK cycle.

Note 4: An absolute maximum delay can be calculated as: (Max # SCK) x (SCK Period) + t_D.

For example, for t_{D1} ($\overline{STB}$ low to $\overline{DTACK}$ low, register write), the # SCK cycles is 2 or 3 and the delay, t_D, is 11.5ns. For a SCK with a 100ns period, the absolute maximum delay is (3 x 100ns) + 11.5, or 311.5ns.

Symbol	Parameter	Conditions	Min	Max	Units
SERIAL SCAN INTERFACE (SSI)					
t_{D5}	Propagation Delay SCK to TCK_SM	*Figure 13*		11.5	ns
t_{D6}	Propagation Delay SCK to TDO_SM	*Figure 13*		12.0	ns
t_{D7}	Propagation Delay SCK to TMS_SM	*Figure 13*		12.5	ns
t_{D8}	Propagation Delay - tpLH SCK to TRIST_SM	*Figure 13*		15.0	ns
t_{D9}	Propagation Delay - tpHL SCK to TRIST_SM	*Figure 13*		12.5	ns
t_{D10}	Propagation Delay SCK to TDO_SM disable	*Figure 13*		12.5	ns
t_{D11}	Propagation Delay SCK to TDO_SM enable	*Figure 13*		14.0	ns
t_{EN1}	Enable Delay $\overline{OE}$ low to TCK_SM, TDO_SM, TMS_SM, or $\overline{TRST0\_SM}$	*Figure 13*		12.0	ns
t_{DIS1}	Disable Delay $\overline{OE}$ high to TCK_SM, TDO_SM, TMS_SM, or $\overline{TRST0\_SM}$	*Figure 13*		11.0	ns
t_{EN2}	Enable Delay $\overline{OE}$ low to TRIST_SM			10.0	ns
t_{DIS2}	Disable Delay $\overline{OE}$ high to TRIST_SM			11.5	ns
t_{DIS3}	Disable Delay $\overline{RST}$ low to $\overline{TRST0\_SM}$			12.5	ns
t_{S2}	Setup Time SCK to TDI_SM	*Figure 13*	3.5		ns
t_{H2}	Hold Time SCK to TDI_SM	*Figure 13*	2.0		ns

Symbol	Parameter	Conditions	Min	Max	Units
TEST & DEBUG INTERFACE TIMING REQUIREMENTS (SCAN)					
t_S	Setup Time TMS to TCK		2.0		ns
t_H	Hold Time TMS to TCK		1.0		ns
t_S	Setup Time TDI to TCK		1.0		ns
t_H	Hold Time TDI to TCK		2.0		ns
t_W	Pulse Width TCK (H or L)		10.0		ns
t_{WL}	Reset Pulse Width $\overline{TRST}$ (L)		2.5		ns
t_{REC}	Recovery Time TCK from $\overline{TRST}$		2.0		ns
f_{MAX}	Maximum Clock Frequency, TCK			25	MHz

Applications/Programmers Reference

TABLE 3. Register Summary

Address	Type	Mnemonic	Register	Active Register Bits	Reset Value
00h	RW	START	Start Register	5	0000h
01h	RW	STATUS	Status Register	10	0800h
02h	RW	INTCTRL	Interrupt Control Register	8	0000h
03h	RW	INTSTAT	Interrupt Status Register	8	0000h
04h	RW	SETUPR	Setup Register	8	0043h
05h	RW	CLKDIV	Clock Divider Register	6	0000h
07h	RW	EXPR	TDI_SM LFSR Exponent Register	3	0000h
08h	RW	LSSEDR	TDI_SM LSB Seed Register	16	0000h
09h	RW	MSSEDR	TDI_SM MSB Seed Register	16	0000h
0Ah	RW	LSRESR	TDI_SM LSB Result Register	16	0000h
0Bh	RW	MSRESR	TDI_SM MSB Result Register	16	0000h
0Ch	RW	INDEXR	Index Register	16	0000h
11h	RW	VINDEXR	Vector Index Register	16	0000h
13h	RW	HTINDEXR	Header/Trailer Index Register	16	0000h
15h	RW	MINDEXR	Macro Index Register	16	0000h
17h	RW	SINDEXR	Sequencer Index Register	16	0000h
19h	RW	BSINDEXR	Bridge Support Register	16	0000h

TABLE 4. Memory/Register Address Map

A4	A3	A2	A1	A0	Function	Base Address	Long Word Index	Structure/Size
0	0	0	0	0	Start	N/A	N/A	16-bit Register
0	0	0	0	1	Status	N/A	N/A	16-bit Register
0	0	0	1	0	Interrput Control	N/A	N/A	16-bit Register
0	0	0	1	1	Interrupt Status	N/A	N/A	16-bit Register
0	0	1	0	0	Setup	N/A	N/A	16-bit Register
0	0	1	0	1	Clock Divider	N/A	N/A	16-bit Register
0	0	1	1	1	TDI_SM LFSR Exponent	N/A	N/A	16-bit Register
0	1	0	0	0	TDI_SM LFSR LSB Seed	N/A	N/A	16-bit Register (Note 5)
0	1	0	0	1	TDI_SM LFSR MSB Seed	N/A	N/A	16-bit Register (Note 5)
0	1	0	1	0	TDI_SM LFSR LSB Result	N/A	N/A	16-bit Register (Note 5)
0	1	0	1	1	TDI_SM LFSR MSB Result	N/A	N/A	16-bit Register (Note 5)
0	1	1	0	0	Index Register	N/A	N/A	16-bit Register (Note 6)
0	1	1	0	1	TDO_SM	0	0 - 0x1BF	(Note 7)
0	1	1	1	0	TDI_SM	0 x 1C0	0 - 0x1BF	(Note 7)
0	1	1	1	1	Expected	0 x 380	0 - 0x1BF	(Note 7)
1	0	0	0	0	Mask	0 x 540	0 - 0x1BF	(Note 7)
1	0	0	0	1	Vector Index	N/A	N/A	16-bit Register
1	0	0	1	0	Vector 1 Vector 2 Vector 3 Vector 4	0 x 700 0 x 700 0 x 700 0 x 700	0x0 - 0x1 0x2 - 0x3 0x4 - 0x5 0x6 - 0x7	(Note 8) *Table 5*
1	0	0	1	1	Header/Trailer Index	N/A	N/A	16-bit Register
1	0	1	0	0	Data Header Data Trailer Instruction Header Instruction Trailer	0 x 708 0 x 728 0 x 748 0 x 768	0x0 - 0x1F 0x20 - 0x3F 0x40 - 0x5F 0x60 - 0x7F	*Table 6*

A4	A3	A2	A1	A0	Function	Base Address	Long Word Index	Structure/Size
1	0	1	0	1	Macro Index	N/A	N/A	16-bit Register
1	0	1	1	0	Macro 1 Macro 2 Macro 3 . . . Macro 16	0 x 788 0 x 789 0 x 78A . . . 0 x 797	0x0 0x1 0x2 . . . 0xF	*Tables 7*, *8*, *9*
1	0	1	1	1	Sequencer Index	N/A	N/A	16-bit Register
1	1	0	0	0	Sequencer	0 x 798	0x0 - 0x1F	*Table 10*
1	1	0	0	1	Scan Bridge Support Index	N/A	N/A	16-bit Register
1	1	0	1	0	Scan Bridge Support	0 x 7B8	0x0 - 0x3F	*Table 11*

Note 5: The TDI_SM LFSR result and seed registers require two sequential reads/writes for each register pair.

Note 6: The index register is used to set the individual address pointers. Writine to the index register will set each of the individual address pointers (TDO_SM, TDI_SM, Expected, and Mask). The individual address pointers will automatically increment with each long word read from TDI_SM or each long word written to the TDO_SM, Expected, or Mask memory spaces.

Note 7: The actual address is calculated from the base address of the memory area plus the content of its address pointer.

Note 8: The upper two bytes of each vector is ignored. These have been inserted to make the space align on long word boundaries.

TABLE 5. Vector Structure

Bit(s)	Function
0x00 - 0x1F	Length (maximum of 4G)
0x20 - 0x27	Macro Number (1 of 256) Room for scaleability
0x28 - 0x2E	Reserved
0x2F	Preloaded data / Load-on-the-fly (LotF)
0x30 - 0x3F	Reserved

TABLE 6. Header/Trailer Structure

Bit(s)	Function
0x00 - 0x1F	32-bit count (Note 9)
0x20 - 0x3FF	124 bytes (992 bits) header/trailer data

Note 9: Count must be greater than zero if the Header/Trailer Usage bits are not equal to "000" or "111".

TABLE 7. Macro Structure

Bit(s)	Function
0x1F	Compare
0x1E	Use Mask / Compare full length of vector (not including header/trailer)
0x1D - 0x1B	Post-shift TCK_SM Count
0x1A - 0x18	Pre-shift TCK_SM Count
0x17	Sync Bit Support Enable
0x16	Macro Structure Bit 8 Enable (Ignored for the shift macros with or without capture)
0x15	Macro Structure bit 7 Enable (Ignored for the shift macros with or without capture)
0x14 - 0x12	Header/Trailer Usage
0x11	Macro Type bit 1
0x10	Macro Type bit 0
0xF - 0x9	Last 7 TMS_SM bits
0x8	Presented during the falling edge of TCK_SM at terminal count during a Shift macro. Use in the same manner as other TMS bits for State and BIST Macros.
0x7	Loop Bit if Macro type is Shift (for 1149.1 it would be a 0) or BIST
0x6 - 0x0	First 7 TMS_SM Bits (LSB is first bit to be shifted out of TMS_SM)

TABLE 8. Header / Trailer Usage

Bit 2	Bit 1	Bit 0	Function
0	0	0	Ignore Headers and Trailers
0	0	1	Use Instruction Header
0	1	0	Use Instruction Trailer
0	1	1	Use both Instruction Header and Trailer
1	0	0	Use Data Header
1	0	1	Use Data Trailer
1	1	0	Use both Data Header and Trailer
1	1	1	Reserved

TABLE 9. Macro Type bits 10 and 11

Bit 1	Bit 0	Function	Function
0	0	BIST Macro	Loop on loop bit for Vector count. No Data
0	1	Shift Macro	Loop on loop bit for vector count. Read data from TDO_SM memory
1	0	Shift Macro with Capture	Loop on loop bit for vector count. Read data from TDO_SM memory
1	1	State Macro	Do not loop on loop bit of macro. No data to be shifted

TABLE 10. Sequencer Structure

Bit(s)	Function
0x00 - 0x1F	Sequence repeat count (up to 255)
0x20 - 0x2F	Vector repeat count
0x30 - 0x3F	Vector number
..x.. - ..x..	Repeat vector repeat count and vector number
0x3E0 - 0x3EF	Vector repeat count (up to 255)
0x3F0 - 0x3FF	Vector number (up to 63)

TABLE 11. Scan Bridge Support Structure

Bit(s)	Function
0x00 - 0x0F	Levels of Scan Bridge support to be inserted in the scan chain
0x10 - 0x17	Hierarchical Level 0 Scan Bridge Address
0x18 - 0x1F	Hierarchical Level 0 Scan Bridge LSP
0x20 - 0x27	Hierarchical Level 1 Scan Bridge Address
0x28 - 0x2F	Hierarchical Level 1 Scan Bridge LSP
..x.. - ..x..	Hierarchical Level Scan Bridge Address and LSP
0x7F0 - 0x7F7	Hierarchical Level 125 Scan Bridge Address
0x7F8 - 0x7FF	Hierarchical Level 125 Scan Bridge LSP

Module Descriptions

Figure 1 shows a high level view of the STA101 which is composed of two main modules, the Parallel Processor Interface (PPI) and the Serial Scan Interface (SSI) which interface to each other through a dual-port memory. The PPI provides a parallel interface for transferring data into and out of the dual-port memory, and for configuring, controlling and obtaining the status of the device. The SSI which resides on the other side of the dual-port memory provides the parallel-to-serial and serial-to-parallel conversion paths for providing test data and test control to support the ScanMaster and IEEE 1532 functions.

Dual Port Memory

The dual port memory will be treated as a separate module in the design to facilitate portability of the RTL of the design to an FPGA host. The Dual Port Memory module is a 2048 x 32 bit dual-port memory which acts as the buffer between the PPI and the SSI. There are seven regions of memory as viewed from the processor side. These regions, shown in *Table 4*, are TDO_SM, TDI_SM, Expected, Mask, Vector, Header/Trailer, Macro. Sequencer, and ScanBridge Support. Each has a pointer which resides in the PPI.

The memory is big endian oriented and is viewed as a single entity from the SSI side and the SSI maintains a pointer. The dual port memory module does not include any logic outside of its own macro function, so all the timing and support logic is included in the following PPI and SSI sections. There will be no logic included in the STA101 design to utilize the "busy"

indicators to keep the user from overwriting memory locations. The only area were this could occur in memory would be the TDI_SM memory space since both the SSI and PPI can write to this space, but the drivers shouldn't allow PPI writes to this area during normal operations.

Parallel Processor Interface

The overall function of the PPI is to receive the parallel data from the processor interface, store the data in its appropriate register or memory location, act on the data if the data are PPI control data, provide status data back to the processor and to provide a read path for result data to the processor. To perform these functions, the PPI consists of seven main blocks of logic along with the dual-port memory. These blocks include the Edge Detector (ED), Processor Interface Controller (PIC), the Memory/ Register Decoder (MRD), the Word/Long Word Converter (WLWC), the Control Generator (CG), the Status/Interrupt Generator (SIG) and the Flag Generator (FG).

WORD/LONG WORD CONVERTER

The Word/Long Word Converter (WLWC) has four 16-bit capture registers, and least significant/ most significant (LS/MS) word read capture register pair and an LS/MS word write capture register pair. Each register within the write register pair has a separate enable to allow for the necessary control to accomplish word to long word conversions when in the 16-bit mode. In 32-bit mode, these enables will be driven simultaneously. A mux is provided in front of the MS word register for the write capture to select between the 32-bit and 16-bit mode external bus. Only one enable and a mux select is needed to control the read capture register pair to accomplish the long word to word conversions when in the 16- bit mode. In the 32-bit mode, the mux selection doesn't change so 32-bits are always driven. A mux is on either side of the LS word register for the read capture. The one at the register output provides for selection between the 32-bit and16-bit mode. The one at the register input is for selection between register space and memory space. All the control for this block is provided by the PIC and MRD with the 16/32 bit mode enable coming from the Setup register.

EDGE DETECTOR

The PPI module can support either an asynchronous or synchronous processor interface. For an asynchronous interface the circuit initially synchronizes $\overline{STB}$ and $\overline{CE}$ to the system clock, SCK, by pipelining these two signals through two flip-flop stages and then performs an edge detection on $\overline{STB}$ and $\overline{CE}$. For a synchronous parallel processor interface this circuit just performs an edge detection. The outputs of this circuit, one clock wide pulses indicating the detection of negative and positive edges, will be used by the Processor Interface Controller (PIC) state machine to start and to end a processor access.

PROCESSOR INTERFACE CONTROLLER

The Processor Interface Controller (PIC) monitors the incoming processor control signals and sets up the appropriate internal control signals to move the data into memory or an internal register on a write or to move the data out of memory or out of an internal register on a read. The PIC edge detects the $\overline{CE}$ and the $\overline{STB}$ to start the access. The PIC provides the control for the word to long word conversion in the WLWC by controlling the three enables and the mux select (READ_MSW) to the capture registers. The PIC also controls when the internal read/write enable is issued to the memory to complete the read/write operation. Timing for register and memory read and write operations is described in *PPI INTERFACE TIMING*.

MEMORY/REGISTER DECODER

The Memory/Register Decoder (MRD) contains all six index registers (Index, Vector Index, Header/Trailer Index, Macro Index, Sequencer Index and ScanBridge Support Index) and four address registers (TDI_SM Address, TDO_SM Address, Expected Address and Mask Address). In general, both index and address registers are used to maintain pointers to their respective memory spaces. The exception is the Index register which is used to set values in the four address registers, i.e., writing to the Index register sets each of the address registers. The value written to each address register is the sum of its base address and the value written to the Index register (the offset). All index and address registers, with the exception of the Index register, will auto-increment with each access to the corresponding memory space.

The MRD provides the address decode to generate all the control and status register enables for the CG and the SIG. The MRD also provides the mux selects for the register or memory selection for the read capture operation in the WLWC.

CONTROL GENERATOR

The Control Generator has the seven control registers within it. The Start, Interrupt Control, Setup, Clock Divider, TDI_SM LFSR Exponent, TDI_SM LFSR LSB Seed, and TDI_SM LFSR MSB Seed registers are all within this block. The CG will issue a strobe to the SSI when a write has been issued to the Start or Setup registers so the SSI can react to the new control data. The strobe will be derived from edge detecting the enables to the Start or Setup registers. The "new" data to the SSI are the Use Sequencer bit and three Use Vector bits from the Start register, and the TDO Default Value, TRST, ScanBridge Support Initiate/Release, three Sync Bit Length, and two Test Loop-back bits from the Setup register.

STATUS/INTERUPT GENERATOR

The Status/Interrupt Generator has the four status registers plus the logic to generate the interrupts and clear the interrupts on a read. The registers are the Status, Interrupt Status, TDI_SM LSFR LSB Result and TDI_SM LFSR MSB Result registers. The SIG receives the LFSR result and strobe signal SSI_LFSR_EN from the SSI and captures the data in the LSB and MSB registers. The SIG receives the compare result bit value from the SSI along with the compare result bit clear and the compare result bit load.

The SIG receives the 4 memory space flags from the FG along with their associated load and clear signals so these bits may be constantly updated. The half-full, half-empty, full and empty flags will be generated and updated regardless of the states of their respective interrupt enables. The SIG also receives the 4 interrupt enables for the flags. The SIG also receives the sequencer active and 3 vector active signals from the SSI. These will also be updated regardless of the enable state.

If an interrupt enable is set then an interrupt will be generated. If an interrupt occurs at the same time as the interrupt status is being read, then the interrupt will be set after the read is complete. All bits in the Interrupt Status register are cleared when the register is read.

FLAG GENERATOR

The FG takes in the TDI_SM or TDO_SM pointer values from the PPI address pointers, compares them and generates the appropriate flags. If a flag condition has occurred, it is passed

along with the corresponding load enable to set the bit in the status register in the SIG. If the flag condition changes then the clear for the corresponding bit is passed to the SIG to clear the flag. The TDO_SM empty and the TDI_SM full flags are passed to the SSI also. A counter enable is passed from the SSI indicate to the FG when the SSI's pointer value has changed. If a decrement and an increment occur at the same time to either of the counters, the counter value will not change.

PPI INTERFACE TIMING

The processor accesses to SCANSTA101 can be classified into six categories:

- register read
- register write
- 16-bit memory read
- 16-bit memory write
- 32-bit memory read
- 32-bit memory write

Register reads and register writes are performed the same whether the device is in 16-bit mode or 32-bit mode. In 32-bit mode, only the LS word is used. The MS word is ignored. All timing for the 16-bit and 32-bit modes are exactly the same.

The 16-bit mode memory write is accomplished by performing two consecutive register writes with the only difference being that the actual write occurs on the second access. The 16-bit mode register read consists of two accesses, with the first access performed similar to the 16-bit register read but requiring one more clock to complete the memory access. Since all 32-bits of the memory data are captured on the first access, the second memory read access is 2 clocks shorter than the first.

The processor initiates a write cycle by asserting $\overline{CE}$ followed by $\overline{STB}$. A set time prior to asserting $\overline{STB}$, the R/$\overline{W}$ is driven low and the address and data buses are driven by valid address and data, respectively. After edge detecting the $\overline{STB}$ and registering all the inputs, the address is decoded to determine which internal address within the STA101 will be written by the processor. The $\overline{DTACK}$ will be asserted on the same rising edge of SCK on which the $\overline{STB}$'s negative edge is detected, indicating to the processor that it can deassert the $\overline{STB}$. When the STA101 detects the positive edge of the $\overline{STB}$, it will deassert the $\overline{DTACK}$ indicating to the processor that it can start a new cycle. The processor can start a new cycle by asserting the $\overline{STB}$ and by driving the address and data buses with new address and data.

A read cycle is similar to the write cycle except that the $\overline{DTACK}$ will not be asserted until the selected address location's contents are loaded. So, for a 16-bit register read it takes one more clock than it does for a write cycle.

Accesses to STA101 memory require two consecutive accesses in the 16-bit external bus mode. The memory writes are similar to register writes but the only difference is that processor has to perform two consecutive 16-bit writes to write to the selected memory location. One important note, during a memory read, is that $\overline{DTACK}$ is not asserted until the contents of the memory is loaded into the capture registers. For this reason the first read from the memory requires five clocks which includes the memory access time, while the second read is done in 3 clock cycles.

Serial Scan Interface

The Serial Scan Interface consists of the following units:

- Clock Divider and TCK_SM Control
- TAP Tracker
- Pointer Generator
- Structure (Sequencer/Vector/Macro/ScanBridge) Decoder
- Structure (Sequencer/Vector/Macro/ScanBridge) Control Registers
- Count Generator
- Shifter (TDO_SM/TDI_SM/TMS_SM)
- Comparator
- Expected and Mask Registers
- Serial Scan Interface Controller (SSIC) and ScanBridge Controller

The clock divider unit divides system clock SCK based on the programmable divisor set in the clock divider to generate TCK_SM. The TCK_SM control unit gates TCK_SM if the TDO_SM buffer is empty.

The TAP Tracker unit keeps track of the target's TAP controller state. The purpose of the TAP Tracker is to determine whether the target's TAP controller is in SIR or SDR state, so that the necessary PAD bits are inserted.

The shifter block contains two 32-bit shift registers for TDO_SM and TDI_SM respectively, and a 16-bit shift register for TMS_SM.

The comparator unit compares the serial input on the TDI_SM pin with the expected, data bit by bit, if the compare bit of the Macro Structure is set. However, if compare/mask bit is set, then the comparator unit compares only those bits that are unmasked.

Expected and Mask Registers contain the data fetched from the memory. This data will be used by the comparator to compare the TDI_SM input with the expected data.

The SSIC provides the timing and control signals to synchronize the operation of the various blocks in the SSI. The ScanBridge Controller consists of the control logic to set up the ScanBridge's hierarchy, if the ScanBridge Support Initiate/Release bit is enabled, prior to scanning actual test vectors out of TDO_SM.

CLOCK DIVIDER AND TCK_SM CONTROL

The clock divider will be a binary divider where only one bit of the clock divider register will be set to one at any given time. The implementation will ignore bits 0, and 8-15, so the supported divisors are 2, 4, 8, 16, 32, 64 and 128.

To generate a TCK_SM of frequency SCK/4, the clock divider register should be set to 4 (00000100). This will enable the gate at the output of bit 2 of the counter to generate a clock of SCK divided by 4. If in LotF mode, then the TCK_SM enable from the SSIC will gate TCK_SM when the TDO_SM buffer is empty.

TAP TRACKER

The TAP Tracker consists of a 16-bit register to trace the IEEE Standard 1149.1 state machine. The state machine is one hot encoded and will continuously track the target's TAP Controller based on the TMS_SM sequence. The TAP Tracker will be used by the ScanBridge support controller to determine whether the target's TAP controller is in SIR or SDR state so that it can insert an appropriate number of pre and post-PAD bits.

The TAP Tracker will enter Test-Logic Reset state upon setting the TRST bit (bit 5) in the Setup register or by issuing a sequence of five TMS_SM high bits.

SHIFTER

The Shifter block contains two 32-bit shift registers for TDO_SM and TDI_SM respectively, and one 16-bit shift register for TMS_SM. The TMS_SM shifter block diagram is shown in *Figure 2*, the TDO_SM shifter block diagram is shown in *Figure 3*, and the TDI_SM shifter block diagram is shown in *Figure 4*.

Before the start of a vector processing the TMS_SM shifter is loaded with the least significant 16 bits of the macro structure. Based on the pre-shift TCK_SM count, the TMS_SM shifter will skip (7 - pre-shift count) least significant bits. e.g., if the pre-shift count is 4, the least significant 3 bits of the TMS_SM shifter will not be used to drive TMS_SM during pre-shift. Similarly, if the post-shift is less than 7 then, during post shift only the number of bits equal to the post-shift count following the macro structure bit 8 will be used to drive TMS_SM.

The STA101 memory is organized in big Endian format. Since a memory write can be accomplished by two consecutive writes to the same location when embedded software loads the TDO_SM memory, it is assumed that the least significant 16 bits are written first and then the most significant 16 bits. Therefore, when the Sequencer or a Vector is initialized the SSIC can directly fetch and load the long word to the TDO_SM shifter without any modification.

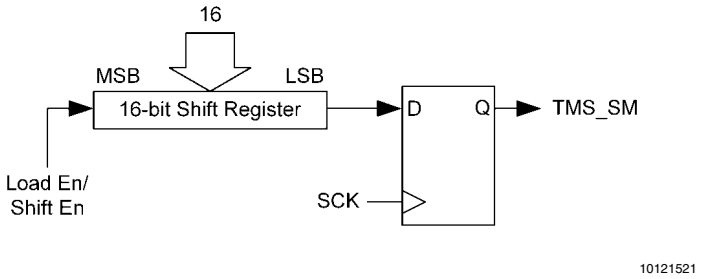


FIGURE 2. TMS_SM Shifter

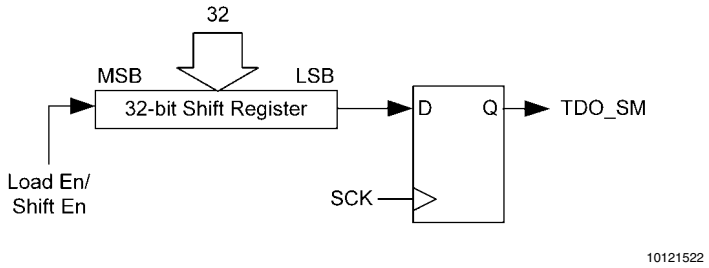


FIGURE 3. TDO_SM Shifter

Similarly, reading from TDI_SM memory can be accomplished by two consecutive reads. When reading from the TDI_SM memory, the first read will contain the least significant 16 bits and the second read the most significant 16 bits.

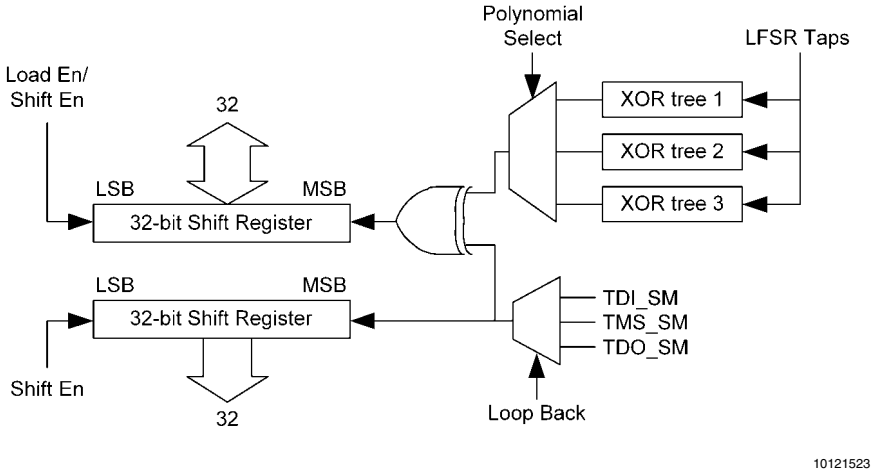


FIGURE 4. TDI_SM Shifter

The TDI_SM shifter unit consists of two 32-bit shift registers as shown in *Figure 4*. The shift register on top will be used as an LFSR register. However, before using the TDI_SM LFSR register, the LFSR Exponent and LFSR Seed registers must be written with valid data. The LFSR Exponent register must be written with a 3-bit binary encoded value such that the corresponding polynomial out of the five available polynomials will be selected. The value written to the LFSR Seed registers will be used to initialize the TDI_SM LFSR register to a predetermined state. Once the test vector has completely scanned in, the final contents of the LFSR register will be transferred to the LFSR Result registers. The 32-bit shift register at the bottom will be used to shift in TDI_SM directly in normal mode or to shift in TMS_SM or TDO_SM in the loopback mode. After shifting in every thirty two bits, the contents of this register will be transferred to the corresponding TDI memory location before the next shift operation.

SHIFTER IMPLEMENTATION

Shift register implementation is illustrated in *Figure 5*. Shift out enable for the TMS_SM and TDO_SM shifters is generated by comparing the clock pulse counter output to the clock divider - 1. Shift in enable for the TDI_SM shifter is generated by comparing the clock pulse counter to programmable divisor/2 - 1. These enables are gated by the control signals from SSIC so that data are shifted out (TMS_SM/TDO_SM) or shifted in (TDI_SM) only when necessary.

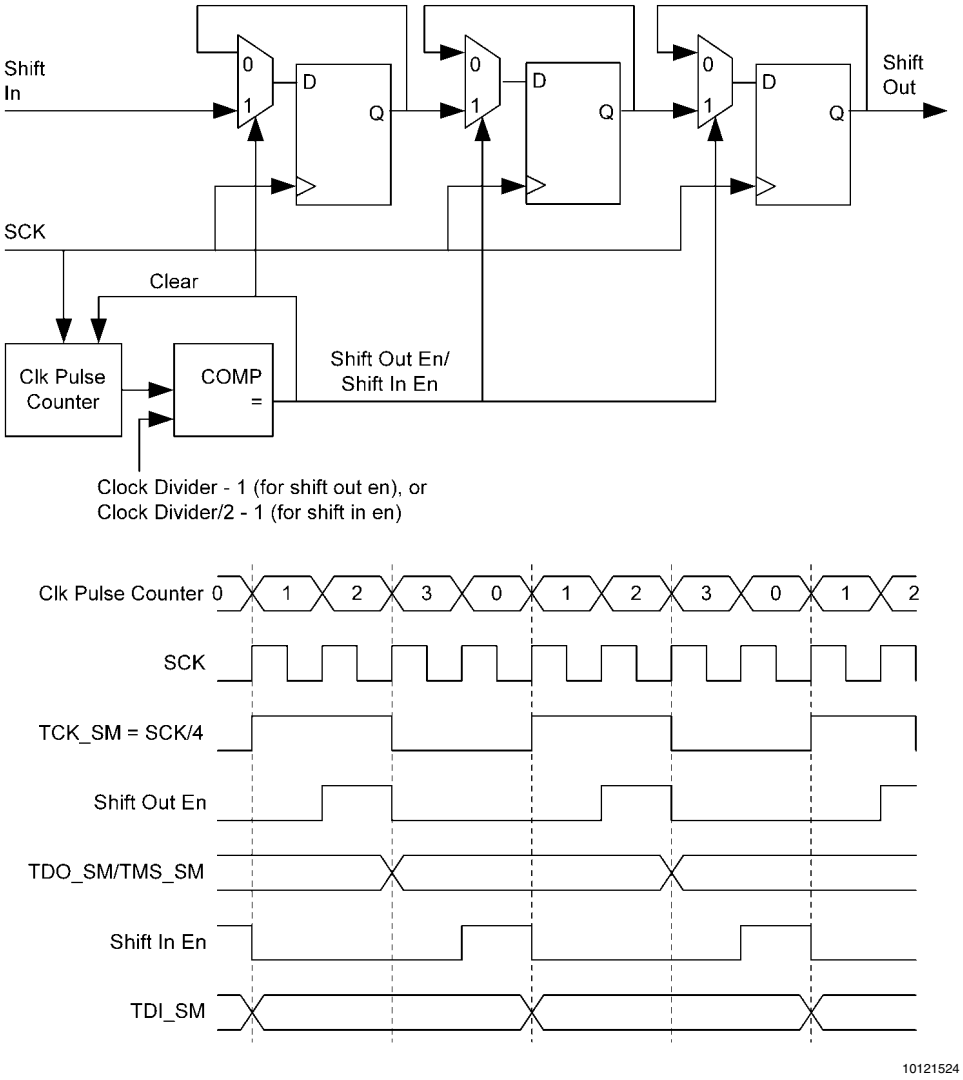


FIGURE 5. Shift Register Implementation and Timing

COMPARATOR AND EXPECTED/MASK REGISTERS

The One Bit Comparator, when enabled, will compare TDI_SM input with expected data. When the compare feature is enabled (pre-load only) the SSIC pre-fetches data into Expected and Mask registers from the address locations pertaining to the current vector being processed. The comparator will compare each bit on the TDI_SM input with the corresponding bit from the expected register. If the mask feature is enabled, then the comparison is performed only on those bits that are not masked, i.e., on those bits whose mask is set to zero. *Table 12* shows how Compare and Use Mask/Compare bits in the Macro Structure will be used.

TABLE 12. Compare and Use Mask/Compare Bit Descriptions

Compare	Use Mask/Compare	Description
0	0	Do Not Compare
0	1	Compare with Mask
1	0	Compare without Mask
1	1	Compare with Mask

Results of Compare bit (bit 15 of Status register) stores the comparison results in the status register. This bit defaults to fail (zero) and will be updated only after the current vector is processed. In the case of a single vector the Results of Compare bit will be set to one (pass) only if all the bits in the scanned in vector match the expected vector. However, in the case of the sequencer only the results of final vector comparison will be taken into account.

Each vector within the sequencer is repeated until the vector repeat count is exhausted. However, the sequence is repeated until either the sequencer repeat count is exhausted or the compare passes and that the loop of the sequence is completed.

Figure 6 illustrates the compare logic.

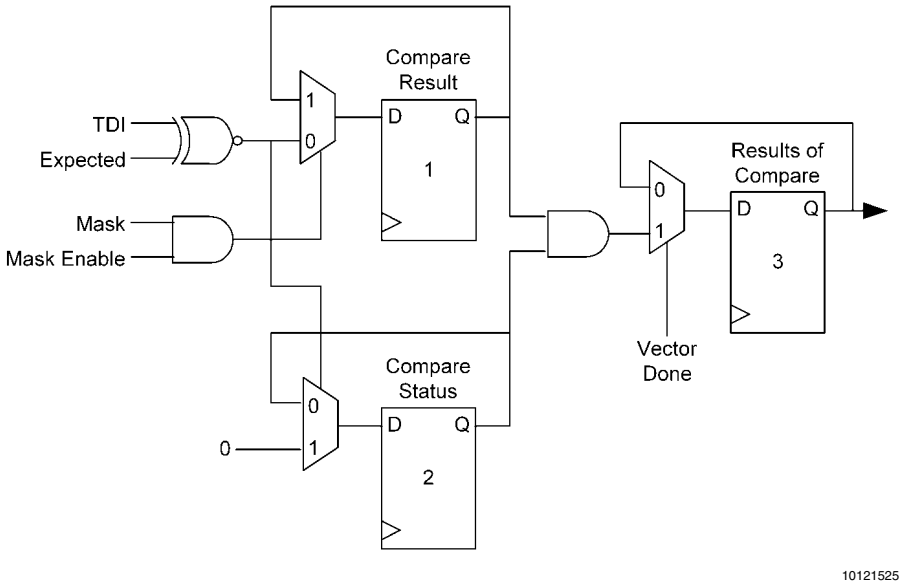


FIGURE 6. Compare Logic

After reset and before every sequencer process, flip-flops 1 and 3 are initialized to zero while flip-flop 2 is set to 1. When the compare feature is enabled flip-flop 1 is continuously updated with the immediate comparison results (1 for pass and 0 for fail). Flip-flop 2 is reset to zero when a mismatch occurs and remains in this state for the remainder of the current vector processing. When the current vector is completely processed flip-flop 3 (Results of Compare register) will be updated with the current status.

SERIAL SCAN INTERFACE CONTROLLER AND SCANBRIDGE CONTROLLER

The Serial Scan Interface Controller (SSIC) remains in the Idle state until new data are written to the Start register. When this event occurs the following operations are performed:

1. If the ScanBridge Support Initiate/Release bit was not set previously and is currently set in the Setup register, the SSIC initializes the ScanBridge Controller (SBC) to perform the following steps to set up all ScanBridges in the hierarchy.
    1. Determine the number of levels of ScanBridge support to be inserted (from the ScanBridge support structure)
2. Sequence TMS_SM so that all ScanBridges in the same level of hierarchy enter the SIR state, and then shift in the address (from the ScanBridge structure) to select a ScanBridge in the current level of hierarchy. The ScanBridge's TAP controller is then sequenced through the Update-IR state.
3. Sequence TMS_SM so that the selected ScanBridge's TAP controller enters the SIR state, then scan in the MODESEL instruction to put its mode register in the data path.
4. Sequence the selected ScanBridge's TAP controller to enter the Shift-DR state and scan in the LSP contents (from the ScanBridge structure) into its mode register. The ScanBridge's TAP controller is then sequenced through the Update-DR state.
5. Repeat Step 1c, but this time scan in the UNPARK instruction so that the LSP is inserted into the active scan chain.
6. Sequence the ScanBridge's TAP controller to enter the RTI state (the LSP will not be unparked until its TAP controller enters RTI).

7. Repeat Steps 1b through 1g to configure the ScanBridges in the remaining hierarchy levels. One set of pre-PAD and post-PAD bits is added to the patterns for each hierarchy level between the ScanMaster and the ScanBridge being configured. The PAD bits are used to bypass the intermediate levels of hierarchy.
8. For the subsequent vectors, if the TAP Tracker enters the
    1. SDR state, the STA101 will add one pre-bit for the PAD register and one post-bit for the bypass register for each level of hierarchy.
    2. SIR state, the STA101 will add one pre-bit for the PAD register and eight post-bits for the ScanBridge instruction register for each level of hierarchy. The eight post-bits will be all ones because the ScanBridge will be forced into bypass mode.
9. The PAD bits need to be stripped when loading a vector into TDI_SM. This will be done by having a status flag to indicate whether the vector that is being scanned out has ScanBridge support or not. If the scanned-out vector has ScanBridge support, then the PAD bits will be stripped when the TAP Tracker enters the SDR or SIR states.

2. If the ScanBridge Support Initiate/Release bit was set previously and is currently reset in the Setup register, the SSIC will toggle TCK_SM five times while TMS_SM is held high. This will return all selected ScanBridges to the wait-for-address state and park the LSPs in the Test-Logic-Reset state. When the ScanBridge support is released the user should make sure that the Use Vector and Use Sequencer bits in the Start register are not set, such that, the SSIC will not start processing a vector or the sequencer immediately after releasing the ScanBridge support. However, once the ScanBridge support is released the user may start processing a vector or the sequencer by writing to the Start register.
3. If the sequencer is enabled (the Use Sequencer bit in the Start register is one),
    1. Clear the Results of Compare bit and set the Using Sequencer bit in the Status register.
    2. Fetch the sequence repeat count.
    3. If the sequence repeat count is zero, the sequence is complete so reset the Using Sequencer bit and return to the Idle state, otherwise fetch the next vector number and its repeat count.
    4. If the vector number is zero, decrement the sequence repeat count and return to Step 3c. If the vector number is illegal, i.e., other than 001, 010, 011, or 100, decrement the sequence repeat count and return to Step 3c.
    5. If the vector repeat count is equal to zero, fetch the next vector number and its repeat count and go to Step 3d. If the repeat count is non-zero fetch the vector structure.
    6. If the pre-load bit in the vector structure is not set, reset the Using Sequencer bit and return to the Idle state.
4. If the sequencer is not enabled but a vector is enabled (the Use Vector bits in the Start register are non-zero), fetch the current vector structure and set the appropriate Using Vector bits in the Status register. If neither the sequencer nor a vector is enabled, return to the Idle state.
5. Fetch the Macro Structure to be used, set the vector/macro control bits and store the TMS_SM bits in the Structure Control registers.
6. If the Pre-shift TCK_SM Count is not zero, then enable TCK_SM and drive TMS_SM using the first seven bits of the macro until the Pre-shift TCK_SM Count is zero. During pre-shift, TDO_SM will be driven with it's previous value.
7. If the macro type is State then,
    1. If the Macro Structure Bit 7 is enabled, set TMS_SM to the bit 7 value of the macro structure and drive TDO_SM with it's previous value.
    2. If the Macro Structure Bit 8 is enabled, set TMS_SM to the bit 8 value of the macro structure and drive TDO_SM with it's previous value and then go to Step 10.
    3. If the sequencer is being used then, decrement the vector repeat count and return to Step 3e. If a vector is being used, return to the Idle state.
8. If the macro type is BIST then,
    1. If the Macro Structure Bit 7 is enabled, set the count length, set TMS_SM to the bit 7 value of the macro structure and drive TDO_SM with the default value (Setup register bit 6) until the count length is zero.
    2. If the Macro Structure Bit 8 is enabled, set TMS_SM to the bit 8 value of the macro structure and drive TDO_SM with the default value (Setup register bit 6) and then go to Step 10.
    3. If the sequencer is being used then, decrement the vector repeat count and return to Step 3e. If a vector is being used, return to the Idle state.
9. If the macro type is Shift or Shift with Capture then,
    1. If the macro type is Shift with Capture, enable TDI capture.
    2. If the Sync Bit Support Enable bit is set, fetch sync bit count, set the count length, set TMS_SM to the loop bit and drive the TDO_SM high until sync bit count is zero.
    3. If the ScanBridge Support Initiate/Release bit is set, drive the TDO_SM with pre- PAD bit (high) and while TMS_SM remains set to the loop bit. Repeat for each level of hierarchy.
    4. If the Use Data/Instruction Header is enabled, fetch the header length and data, set the count length, and drive the TDO_SM with header data until the header length is zero and while TMS_SM remains set to the loop bit.
    5. If the Compare or Mask/Compare is set, enable the comparator.
    6. Set the vector count length, and drive the TDO_SM with vector data until the count length is one and while TMS_SM remains set to the loop bit. In the LotF mode if the count length is not zero and the TDO buffer is empty, then gate TCK_SM until more data are available in the TDO buffer. When TCK_SM is disabled TMS_SM and TDO_SM will be driven with their previous values.
    7. If the Use Data/Instruction Trailer is enabled, fetch the trailer length and data, set the count length, and drive TDO_SM with trailer data until the trailer length is one and while TMS_SM remains set to the loop bit.
    8. If the ScanBridge Support Initiate/Release bit is set:

1. If the TAP tracker is in the Shift-IR state and the number of levels of hierarchy is greater than one, set the count length to eight, and drive TDO_SM with post-PAD bits (all high) until the count length is zero for each level of hierarchy and while TMS_SM remains set to the loop bit.
2. If the TAP tracker is in the Shift-DR state and the number of levels of hierarchy is greater than one, drive TDO_SM with a post-PAD bit (high) for each level of hierarchy and while TMS_SM remains set to the loop bit.
3. For the final level of hierarchy or if there is only one level of hierarchy, and if the TAP tracker is in the Shift-IR state, set the count length to eight, and drive TDO_SM with post-PAD bits (all high) until the count length is one and while TMS_SM remains set to the loop bit.

9. If the Sync Bit Support Enable is set, fetch sync bit count, set the count length, and drive the TDO_SM high until sync bit count is one and while TMS_SM remains set to the loop bit.
10. Set TMS_SM to the bit 8 of the TMS_SM Macro Structure sequence and drive TDO_SM with the final vector bit or trailer bit or post-PAD bit or sync bit. After shifting out the final vector bit, disable the comparator and register the comparison results.
10. If the Post-shift TCK_SM Count is not zero, then enable TCK_SM and drive TMS_SM using the last seven bits of the macro until the Post-shift TCK_SM Count is zero.
11. If the Sequencer is being used,
    1. Decrement the sequence repeat count and return to Step 3c if the Compare or Mask/Compare is enabled and the results of compare is a fail.
    2. Decrement the vector repeat count and return to Step 3e if the if the Compare or Mask/Compare is enabled and the results of compare is a pass.
    3. Decrement the vector repeat count and return to Step 3e if the Compare or Mask/ Compare is not enabled.
12. If the Vector is being used return to the Idle state.

MODE REGISTER WRITE TO VECTOR/SEQUENCER START

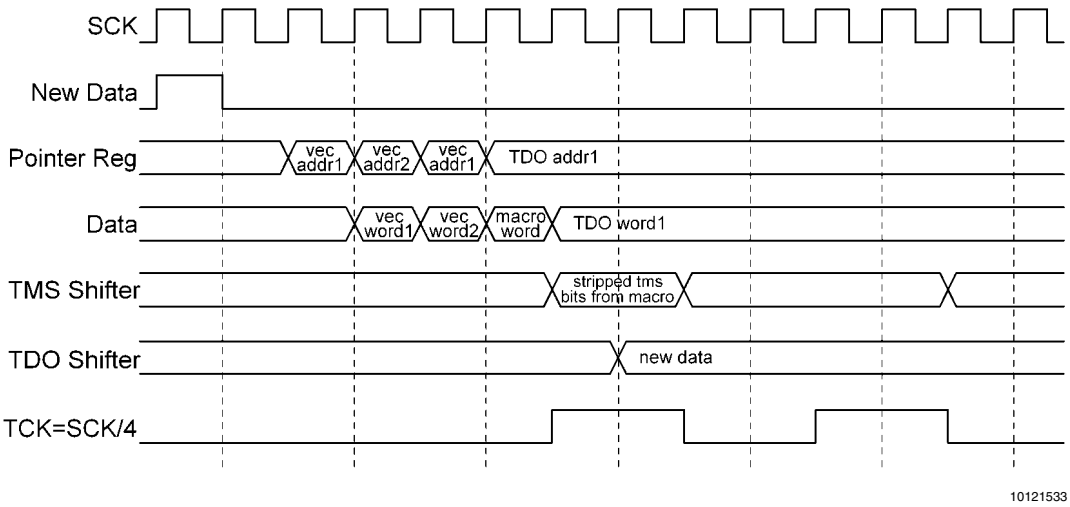


FIGURE 7. Timing from Mode Register Write to Vector Start

Figure 7 shows the timing from the processor write to the start of vector processing, whereas *Figure 8* shows the timing from the processor write to the start of sequencer processing. A processor write to the Start registers is indicated by a "new data" pulse. On the same SCK rising edge when the "new data" is detected to be high, the Start or Setup register contents will be updated with new data. So, the decoding of the enables takes place during the next clock cycle to determine whether to process the sequencer or a vector. Therefore, one clock after the "new data" is detected, the SSIC starts loading the pointer register on consecutive cycles with the appropriate addresses to fetch the Sequencer, Vector and Macro Structures. Once the headers are decoded and Structure Control Registers are set up, the SSIC loads the pointer register so that data from the TDO_SM memory area is fetched and loaded into the TDO_SM shifter before being shifted out. However if there are any sync bits and/or header bits and/or ScanBridge support is enabled, then the sync bits and/or header bits and/or ScanBridge pre-PAD bits will be loaded into the TDO_SM shifter before processing the actual test vector. Once the actual test vector is completely shifted out, again depending on the ScanBridge support and/or the use of trailers, post-PAD bits and the trailer bits are loaded and shifted out through the TDO_SM shifter.

The count length will be decremented by one with each shift. After shifting out all the current shifter contents the shifter will be loaded with new data before the falling edge of the next TCK_SM, if the count length is not exhausted. In the case where data cannot be loaded from the memory before the next falling edge of TCK_SM, the TCK_SM will be gated until the data is available.

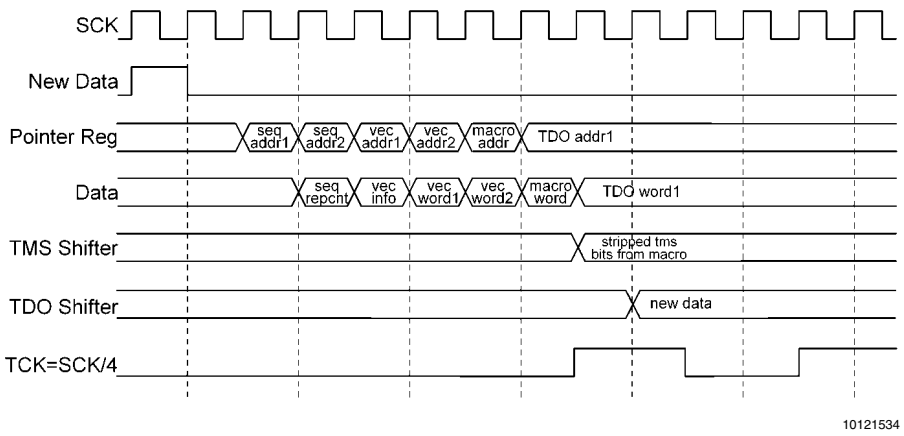


FIGURE 8. Timing from Mode Register to Sequencer Start

WRITING AND READING PARTIAL LONG WORDS

Care should be taken when writing a partial long word to TDO_SM memory or reading a partial long word from TDI_SM memory. Since the TDO_SM shifter shifts out LSB first, the valid (meaningful) bits within a partial long word (i.e., long word containing less than 32 valid bits to be shifted to the scan chain) must be stored and written into the memory as the least significant bits. This will assure that the desired bits will be accurately loaded into the TDO_SM shifter and shifted out to the boundary scan chain. For instance, to shift a 3-bit (110) sequence the partial long word should be written to the TDO_SM memory as shown in

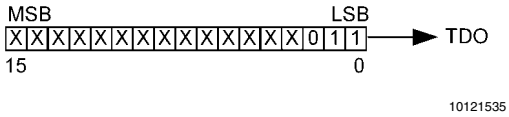


FIGURE 9. Writing a Partial Long Word to the TDO_SM Memory

Figure 9 (only the least significant 16 bits are shown). A subsequent enable and load of the vector structure with the correct length will initialize the shift operation and only the bits that are significant will be shifted out to the scan chain.

Data is shifted from the scan chain into the TDI_SM shifter from MSB to LSB. Consequently, the valid (i.e., meaningful) bits in a partial long word shifted into the TDI_SM shifter will reside in the upper significant bit locations. For example, if a scan operation involves shifting and evaluating 69 bits returning to TDI_SM, the TDI_SM memory will be loaded with two long words (i.e., two full long words plus a partial long word containing 5 meaningful bits). If the last 5 bits shifted back to the TDI_SM shifter are 11010, then upon completion of the shift operation, the TDI_SM shifter will contain the following partial long word as shown in *Figure 10* (only most significant 16 bits are shown), which will subsequently be loaded into the TDI_SM memory.

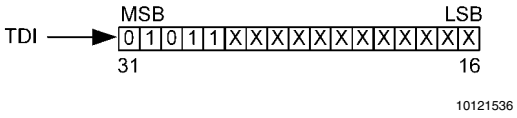


FIGURE 10. Reading a Partial Long Word from the TDI_SM Memory

Following a read of a partial long word, the embedded test software must adjust the position of the valid bits read from the TDI_SM shifter/buffer or the position of the expected data to assure that an accurate comparison is made (and the non-meaningful bits are masked).

TDO_SM IMPLEMENTATION

The behavior of the TDO_SM output depends on the current macro type that is being processed and the SETUP register bits 11 and 10, as shown in *Table 13*, regardless of the TAP tracker state. For shift macros, the TDO_SM output also depends on the current macro structure's TMS_SM bit number as explained below.

TABLE 13. TDO_SM Output Behavior

SETUP[11:10]	TDO_SM
00	Hold Previous value
01 or 10	Default TDO value (Bit 6 of the SETUP register) (Note 10)
11	High Impedance

Note 10: Default TDO value (bit 6 of the SETUP register) may be set to a 0 when SETUP[11:10]=01 and to a 1 when SETUP[11:10]=10.

For BIST and STATE macros, the TDO_SM output behaves exactly as shown in the above table, however, for the shift macros, with or without capture, the TDO_SM output behaves as per the table only when the corresponding TMS_SM output is not driven by the macro structure bit 7 or 8. On each falling edge of the TCK_SM following the TCK_SM's falling edge on which the TMS_SM changes state from bit 6 of the macro structure to the bit 7of the macro structure, the serial test vector data fetched from the memory will be presented on the TDO_SM output. On the falling edge of the TCK_SM on which the final bit of the test vector is presented on the TDO_SM output, the TMS_SM will be presented with the macro structure bit 8. On the consequent TCK_SM falling edges and on the TCK_SM falling edges before the TMS_SM changes state from bit 6 to bit 7 of the macro structure the TDO_SM will behave as per the table above.

Hardware Interface Details

TABLE 14. System Interface Signal Description

Signal Name	No. of Bits	Pin Type	Driver Type	Freq. MHz	Description
SCK	1	I	LVTTL	66	System Clock: This is the main clock signal to the STA101. SCK is used to clock all internal circuitry
$\overline{RST}$	1	I,H	LVTTL	N/A	Hardware Reset signal (with hysteresis (H)): This is the STA101 asynchronous reset signal.This signal resets the entire STA101 and sets all registers to their respective default values.
$\overline{OE}$	1	I	LVTTL	N/A	Output enable: Tristates all dot1 outputs when high.

TABLE 15. Parallel Processor Interface Signal Descriptions

Signal Name	No. of Bits	Pin Type	Driver Type	Freq. MHz	Description
DATA(31:16)	16	I/O	LVTTL (weakest driver)	N/A	Bidirectional Data Bus. Not bonded out in packaged part. These are only used in the 32-bit macro version.
DATA(15:0)	16	I/O	LVTTL	N/A	Bidirectional Data Bus.
ADDRESS(4:0)	5	I	LVTTL	N/A	Address Bus
$\overline{CE}$	1	I	LVTTL	N/A	Chip Enable, when low, enables the PPI for transfers. DATA (31:0) and $\overline{DTACK}$ are tristated when $\overline{CE}$ is high.
R/$\overline{W}$	1	I	LVTTL	N/A	Read/Write defines a PPI cycle. Read when high, write when low.
$\overline{STB}$	1	I	LVTTL	N/A	Strobe is used for timing all PPI transfers. DATA(31:0) are tristated when $\overline{STB}$ is high. Data valid setup is with respect to the falling edge of $\overline{STB}$ and data valid hold is with respect to rising edge of $\overline{STB}$.
$\overline{DTACK}$	1	O	O/D	N/A	Data Acknowledge (open drain - sustained tristate). $\overline{DTACK}$ is used to synchronize asynchronous transfers between the host and the STA101. During write cycles, $\overline{DTACK}$ goes low when data has been registered and then goes to high impedance when the cycle has been completed. During read cycles $\overline{DTACK}$ goes low when data bus is driven with the valid data and then goes to high impedance when the cycle has been completed.
INT	1	O	LVTTL	N/A	Interrupt is used to trigger a host interrupt for any of the defined interrupt events. Signal is active high.

TABLE 16. Serial Scan Interface Signal Descriptions

Signal Name	No.	Pin Type	Driver Type	Freq. MHz	Description
TDI_SM	1	I	LVTTL	up to 25	ScanMaster Test Data Input (weak pullup)
TDO_SM	1	O	LVTTL	up to 25	ScanMaster Test Data Output
TMS_SM	1	O	LVTTL	up to 25	ScanMaster Test Mode Select
TCK_SM	1	O	LVTTL	up to 25	ScanMaster Test Clock
$\overline{\text{TRST0\_SM}}$	1	O	LVTTL	N/A	ScanMaster Test Reset
$\overline{\text{TRST1\_SM}}$	1	O	LVTTL	N/A	Redundant ScanMaster Test Reset (not bonded out)
TRIST_SM	1	O	LVTTL	N/A	The tristate notification pin exerts a high when TDO_SM is tristated.

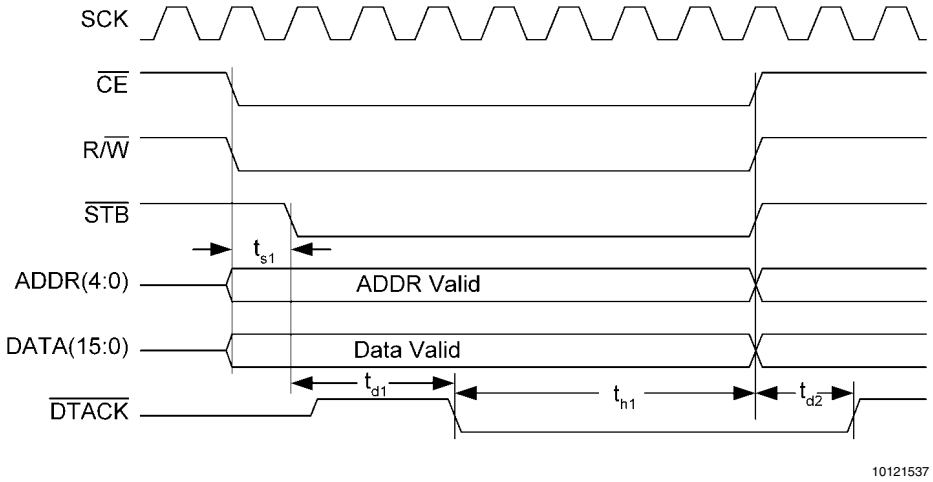


FIGURE 11. PPI Write Cycle Timing Diagram

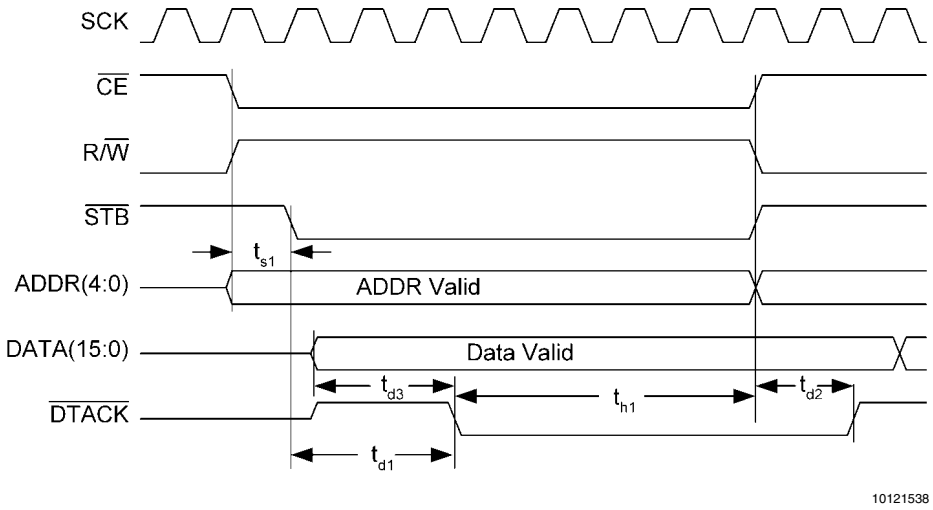


FIGURE 12. PPI Read Cycle Timing Diagram

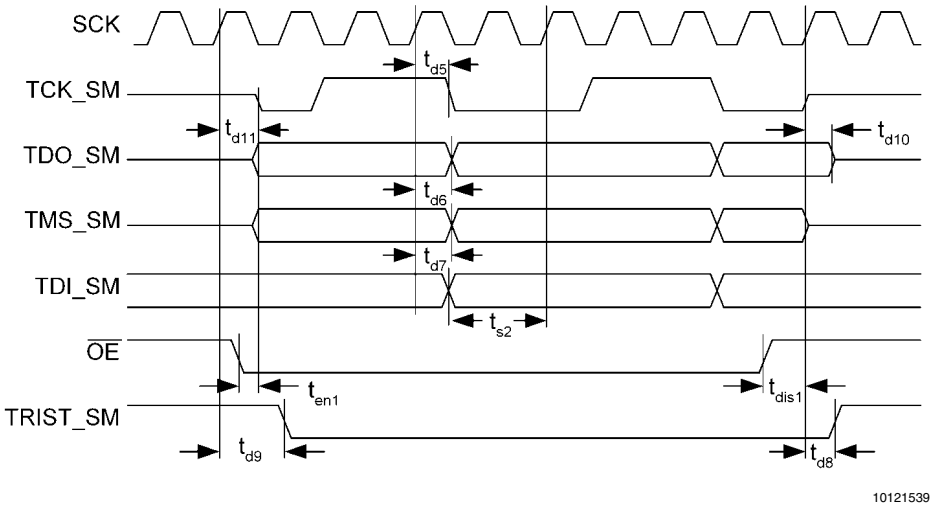


FIGURE 13. SSI Timing Diagram with Clock Divider set to 4

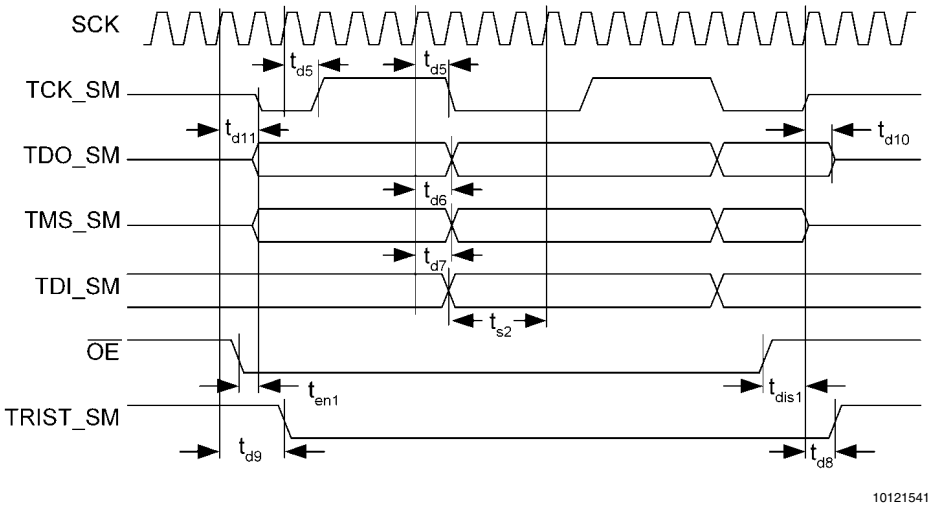


FIGURE 14. SSI Timing Diagram with Clock Divider set to 8

TABLE 17. STA101 1149.1 Signal Descriptions

Signal Name	No. of Bits	Pin Type	Driver Type	Freq. MHz	Description
TDO	1	O	LVTTL	up to 25	STA101 Test Data Out
TDI	1	I,U	LVTTL	up to 25	STA101 Test Data In (pullup (U))
TMS	1	I,U	LVTTL	up to 25	STA101 Test Mode Select (pullup (U))
TCK	1	I	LVTTL	up to 25	STA101 Test Clock
$\overline{TRST}$	1	I,U,H	LVTTL	N/A	STA101 Test Reset (pullup (U) & hysteresis (H))

TABLE 18. STA101 Scan Signal Descriptions

Signal Name	No. of Bits	Pin Type	Driver Type	Freq. MHz	Description
SCAN_EN	1	I	Shared TRIST	TBD	STA101 Scan Enable Shared pin with TRIST.
SCAN_IN	1	I	Shared DATA15	TBD	STA101 Scan Data In. Shared pin with DATA15.
SCAN_OUT	1	O	Shared DATA14	TBD	STA101 Scan Data Out. Shared pin with DATA14.

TEST AND DEBUG INTERFACE

The test and debug interfaces are provided to perform manufacturing tests. There is a standard JTAG interface along with a scan interface. The scan interface have shared pins with the external data pins. Scan is selected by a user defined instruction through the JTAG port. Note that the scan chain (s) will not be hooked up to the JTAG tap.

SAFE MODE

This device implements the following design rules to provide SEU/SEE protection:

- Triple modular redundancy for $\overline{\text{TRST0\_SM}}$ and $\overline{\text{TRST1\_SM}}$ outputs with the help of a TMR D flip-flop .
- After reset all scan interface outputs are driven to SEU tolerant safe values as shown below:
  TMS_SM = 1
  TCK_SM = 0
  TDO_SM = Z
  $\overline{\text{TRST0\_SM}}$ = 0
  $\overline{\text{TRST1\_SM}}$ = 0
- The EXTEST and the HIGHZ outputs from the JTAG TAP controller are gated with $\overline{\text{TRST}}$ to protect the boundary scan cells from inadvertantly entering the test mode.

Software Interface Details

CLOCK GENERATION AND DISTRIBUTION

Input Clock (SCK): Up to 66 MHz

Output Clock (TCK_SM): TCK_SM is a divided, registered version of SCK.

- Selectable: to 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, and 1/128 of SCK.
- Frequency: up to 25 MHz

RESET STRATEGY

The incoming external hardware reset ($\overline{\text{RST}}$) will be synchronized to the incoming clock (SCK) and is combined with the soft reset to generate a synchronized internal reset (SYS_RST_N). During operation, the chip can be reset by writing a '1' to the Reset bit in the Setup register. All logic throughout the device will be initialized, all control and status registers will be in a known default state, all PPI memory address pointers will default to their respective base addresses, the SSI memory pointer will default to zero, the Tap Tracker will be reset to TLR, and the clock division counter will be initialized to all zero's after deassertion of the internal reset. The Reset bit in the Setup register is self clearing. The TRST bit in the Setup register, when set, resets the SSI logic and drives the $\overline{\text{TRST0\_SM}}$ and $\overline{\text{TRST1\_SM}}$ to zero.

REGISTER DEFINITIONS

The following sections include descriptions of each addressable register in the ScanMaster memory space. Following the title of the particular register, the mnemonic for the register is included in parentheses as well as the physical address location in hexadecimal notation (value preceded by $). **KEY**- RO: Read Only; RW: Read/Write.

TABLE 19. Start Register (START) ($00)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15:14	RO	Reserved	0	00b	
13	RW	Onboard Memory BIST	0	0b	$\overline{\text{SYS\_RST}}$
12:9	RO	Reserved	0	0000b	
8	RW	Use Sequencer	0	0b	$\overline{\text{SYS\_RST}}$
7:3	RO	Reserved Use Vector x (Note 11)	0	0000h	
2:0	RW	Use Vector x	0	000b	$\overline{\text{SYS\_RST}}$

Note 11: Reserved Use Vector x for future growth for the number of vectors.

Onboard Memory BIST	ScanMaster memory BIST enable. This bit is self clearing when BIST result is written to the Memory BIST Result bit in the Status register.
'1'	Initiate on chip memory BIST
'0'	On chip memory BIST complete
Use Sequencer	Sequencer enable/disable (For preloaded vectors only)
'1'	Enable sequencer
'0'	Disable sequencer
Use Vector <2:0>	Use Vector x designates the vector "x" which is enabled, where "x" is the vector number, a binary encoding of bits <2:0>. Only vectors 1 through 4 are valid. Vectors 5 through 7 reserved for future use.
'000'	No vector enabled
'001'	Vector 1 enabled
'010'	Vector 2 enabled
'011'	Vector 3 enabled
'100'	Vector 4 enabled

TABLE 20. Status Register (STATUS) ($01) (Note 12)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15	RW	Results of Compare (Note 13)	0	0b	$\overline{\text{SYS\_RST}}$
14	RW	BIST Running	0	0b	$\overline{\text{SYS\_RST}}$
13	RW	Memory BIST Result	0	0b	$\overline{\text{SYS\_RST}}$
12	RW	TDO Status Half-empty (Note 15)	0	0b	$\overline{\text{SYS\_RST}}$
11	RW	TDO Status Empty	0	0b	$\overline{\text{SYS\_RST}}$
10	RW	TDI Status Full	0	0b	$\overline{\text{SYS\_RST}}$
9	RW	TDI Status Half-full (Note 15)	0	0b	$\overline{\text{SYS\_RST}}$
8	RW	Using Sequencer	0	0b	$\overline{\text{SYS\_RST}}$
7:3	RO	Reserved Using Vector x (Note 14)	0	00000b	
2:0	RW	Using Vector x	0	000b	$\overline{\text{SYS\_RST}}$

Note 12: Write capability to the register is only for test and debug purposes. Drivers should disable writes to register during normal operation.

Note 13: Results of Compare bit is toggled after a compare is complete and it is set to the mismatch state when sequencer is kicked off again. Remains in last state until next compare completed or until set to the mismatch.

Note 14: Reserved Using Vector x for future growth for the number of vectors.

Note 15: Half full or half empty designates 56 long words.

Results of Compare	Results of compare between TDI_SM and Expected memory space
'1'	Compare match
'0'	Compare mismatch
BIST Running	Indicates the BIST operation is still active
'1'	BIST operation active
'0'	BIST operation complete or BIST operation not running
Memory BIST Result	ScanMaster BIST result. BIST result will be held until overwritten by next BIST operation.
'1'	Passed memory BIST
'0'	Failed memory BIST
TDO Status Half	TDO_SM memory space status half empty
'1'	TDO_SM memory space half empty
'0'	TDO_SM memory space not half empty
TDO Status Empty	TDO_SM memory space status empty
'1'	TDO_SM memory space empty
'0'	TDO_SM memory space not empty
TDI Status Full	TDI_SM memory space status full
'1'	TDI_SM memory space full
'0'	TDI_SM memory space not full
TDI Status Half	TDI_SM memory space status half full
'1'	TDI_SM memory space half full
'0'	TDI_SM memory space not half full
Using Sequencer	Sequencer active and processing. Bit cleared when sequence complete.
'1'	Sequencer active
'0'	Sequencer inactive
Using Vector<2:0>	Using Vector x designates the vector "x" currently running, where "x" is the vector number, a binary encoding of bits <2:0>. Only vectors 1 through 4 are valid. Vectors 5 through 7 reserved for future use. Field cleared when vectors complete.
'000'	No vector active.
'001'	Vector 1 active.
'010'	Vector 2 active.
'011'	Vector 3 active.
'100'	Vector 4 active.

TABLE 21. Interrupt Control Register (INTCTRL) ($02)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15:13	RO	Reserved	0	000b	
12	RW	TDO Half-empty Interrupt Enable (Note 18)	0	0b	$\overline{\text{SYS\_RST}}$
11	RW	TDO Empty Interrupt Enable	0	0b	$\overline{\text{SYS\_RST}}$
10	RW	TDI Full Interrupt Enable	0	0b	$\overline{\text{SYS\_RST}}$
9	RW	TDI Half-full Interrupt Enable (Note 18)	0	0b	$\overline{\text{SYS\_RST}}$
8	RW	Sequencer Interrupt Enable (Note 17)	0	0b	$\overline{\text{SYS\_RST}}$
7:3	RO	Reserved Vector x Interrupt Enable (Note 16)	0	00000b	
2:0	RW	Vector x Interrupt Enable (Note 17)	0	000b	$\overline{\text{SYS\_RST}}$

Note 16: Reserved Vector x Interrupt Enable for future growth for the number of vectors.

Note 17: Drivers should not allow Sequencer Interrupt Enable and Vector x Interrupt Enable to be set at same time. Sequencer Interrupt Enable has priority over the Vector x Interrupt Enable.

Note 18: Half full or half empty designates 56 long words.

TDO Half-empty Interrupt Enable	TDO_SM memory space half empty interrupt enable
'1'	Enable TDO_SM memory space half empty interrupt
'0'	Disable TDO_SM memory space half empty interrupt
TDO Empty Interrupt Enable	TDO_SM memory space empty interrupt enable
'1'	Enable TDO_SM memory space empty interrupt
'0'	Disable TDO_SM memory space empty interrupt
TDI Full Interrupt Enable	TDI_SM memory space full interrupt enable
'1'	Enable TDI_SM memory space full interrupt
'0'	Disable TDI_SM memory space full interrupt
TDI Half-full Interrupt Enable	TDI_SM memory space half full interrupt enable
'1'	Enable TDI_SM memory space half full interrupt
'0'	Disable TDI_SM memory space half full interrupt
Sequencer Interrupt Enable	Sequencer activity complete interrupt enable
'1'	Enable Sequencer activity complete interrupt
'0'	Disable Sequencer activity complete interrupt
Vector Interrupt Enable<2:0>	Vector "x" complete interrupt enable, where "x" is the vector number, a binary encoding of bits <2:0>. Only vector interrupts1 through 4 are valid. Vector interrupts 5 through 7 reserved for future use.
'000'	No vector interrupt enabled
'001'	Vector 1 interrupt enabled
'010'	Vector 2 interrupt enabled
'011'	Vector 3 interrupt enabled
'100'	Vector 4 interrupt enabled

TABLE 22. Interrupt Status Register (INTSTAT) ($03) (Note 22)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15:13	RO	Reserved	0	00b	
12	RW	TDO Half-empty Interrupt (Note 21)	0	0b	$\overline{\text{SYS\_RST}}$
11	RW	TDO Empty Interrupt	0	1b	$\overline{\text{SYS\_RST}}$
10	RW	TDI Full Interrupt	0	0b	$\overline{\text{SYS\_RST}}$
9	RW	TDI Half-full Interrupt (Note 21)	0	0b	$\overline{\text{SYS\_RST}}$
8	RW	Sequencer Interrupt	0	0b	$\overline{\text{SYS\_RST}}$
7:3	RO	Reserved Vector x Interrupt (Notes 19, 20)	0	00000b	
2:0	RW	Vector x Interrupt (Note 20)	0	000b	$\overline{\text{SYS\_RST}}$

Note 19: Reserved Vector x Interrupt for future growth for the number of vectors.

Note 20: Drivers shouldn't allow Sequencer Interrupt and Vector x Interrupt to be set at same time. Sequencer Interrupt has priority over the Vector x Interrupt.

Note 21: Half full or half empty designates 56 long words.

Note 22: This register is writable in debug mode only.

TDO Half-empty Interrupt	TDO_SM memory space half empty status
'1'	TDO_SM memory space half empty
'0'	TDO_SM memory space not half empty
TDO Empty Interrupt	TDO_SM memory space empty status
'1'	TDO_SM memory space empty
'0'	TDO_SM memory space not empty
TDI Full Interrupt	TDI_SM memory space full status
'1'	TDI_SM memory space full
'0'	TDI_SM memory space not full
TDI Half-full Interrupt	TDI_SM memory space half full status
'1'	TDI_SM memory space half full
'0'	TDI_SM memory space not half full
Sequencer Interrupt	Sequencer completed status
'1'	Sequencer processing completed
'0'	Sequencer processing or not started
Vector Interrupt<2:0>	Vector "x" completed status, where "x" is the vector number, a binary encoding of bits <2:0>.Only vectors 1 through 4 are valid. Vectors 5 through 7 reserved for future use.
'000'	No vector completed activity
'001'	Vector 1 completed
'010'	Vector 2 completed
'011'	Vector 3 completed
'100'	Vector 4 completed

TABLE 23. Setup Register (SETUPR) ($04)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15	RW	16/32 bit Mode	0	0b	$\overline{\text{SYS\_RST}}$
14:10	RO	Reserved	0	00h	
11:10	RW	TDO_SM Ctrl	0	00b	$\overline{\text{SYS\_RST}}$
9:7	RW	Sync Bit Length	0	000b	$\overline{\text{SYS\_RST}}$
6	RW	Default TDO Value	0	1b	$\overline{\text{SYS\_RST}}$
5	RW	Debug Mode	0	0b	$\overline{\text{SYS\_RST}}$
4	RW	ScanBridge Support Initiate/ Release	0	0b	$\overline{\text{SYS\_RST}}$
3	RW	TRST	0	0b	$\overline{\text{SYS\_RST}}$
2	RW	Reset	0	0b	$\overline{\text{SYS\_RST}}$
1:0	RW	Test Loop-Back	0	11b	$\overline{\text{SYS\_RST}}$

16/32 bit Mode	Selects 16-bit or 32-bit external interface mode
'1'	32-bit external interface mode (used in macro form only)
'0'	16-bit external interface mode
TDO_SM Ctrl<11:10>	TDO_SM Control bits
'00'	Hold previous value
'01'	Default TDO Value
'10'	Default TDO Value
'11'	High impedance
Sync Bit Length "x"	Sync Bit Length bits represents the number of sync bits to be used when the Sync Bit Support Enable bit (17 in the Macro Structure) is set. Where "x" is the binary encoded numeric value.
Default TDO Value	The value in this register will be sent out on the TDO_SM pin when performing a BIST or a STATE Macro.
'1'	Drive TDO_SM to one.
'0'	Drive TDO_SM to zero.
Debug Mode	Control bit to put STA101 in debug mode
'1'	Debug mode.
'0'	Normal mode.
ScanBridge Support Initiate/ Release	ScanBridge support enable
'1'	Enable ScanBridge support
'0'	Disable ScanBridge support
TRST	Processor initiated ScanMaster test reset (on $\overline{\text{TRST0\_SM}}$ and TRST1_SM_N). Bit is cleared by a processor write.
'1'	Set TRST outputs low (active) and reset SSI logic.
'0'	Set TRST outputs high
Reset	Processor commanded synchronous reset to the serial scan logic for 2 clocks. This bit is self clearing.
'1'	Reset the entire chip.
'0'	Release serial scan logic reset
Test Loop-Back<1:0>	Test loop-back mode bits
'00'	Normal operation
'01'	Loop-back TDO_SM to TDI_SM
'10'	Loop-back TMS_SM to TDI_SM
'11'	All Dot1 (1149.1) pins placed in SEU tolerant safe mode with: TMS_SM = 1, TCK_SM = 0, TDO_SM = Z, $\overline{\text{TRST0\_SM}}$ = 0

TABLE 24. Clock Divider Register (CLKDIV) ($05)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15:8	RO	Reserved	0	00h	
7:1	RW	Divisor	0	00h	$\overline{\text{SYS\_RST}}$
0	RO	Reserved (hard coded) (Note 23)	0	0b	

Note 23: LSB of the Clock Divider register is hard coded to zero.

Divisor<7:1>	Clock divisor for the division of the SCK clock to the serial scan clock.
'0000000'	No serial scan clock generated.
'0000001'	Divide SCK by 2
'0000010'	Divide SCK by 4
'0000100'	Divide SCK by 8
'0001000'	Divide SCK by 16
'0010000'	Divide SCK by 32
'0100000'	Divide SCK by 64
'1000000'	Divide SCK by 128

TABLE 25. TDI_SM LFSR Exponent Register (EXPR) ($07)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15:3	RO	Reserved	0	0000h	
2:0	RW	LFSR	0	000b	$\overline{\text{SYS\_RST}}$

LFSR Exponent<2:0>	LFSR exponent. Binary encoding for the selection between three polynomials.
'000'	No polynomial selected
'001'	Polynomial 1: $X^{32} + X^7 + X^5 + X^3 + X^2 + X + 1$
'010'	Polynomial 2: $X^{32} + X^{28} + X^{27} + X + 1$
'011'	Polynomial 3: $X^{32} + X^7 + X^6 + X^2 + 1$

TABLE 26. TDI_SM LFSR LSB Seed Register (LSSEDR) ($08) (Notes 24, 25)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15:0	RW	LSW LFSR Seed	0	0000h	$\overline{\text{SYS\_RST}}$

Note 24: LSW LFSR Seed<15:0> is the LS word of the LFSR seed.

Note 25: This register along with register MSSEDR form a register pair and should be read/written with two consecutive read/write accesses.

TABLE 27. TDI_SM LFSR MSB Seed Register (MSSEDR) ($09) (Notes 26, 27)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15:0	RW	MSW LFSR Seed	0	0000h	$\overline{\text{SYS\_RST}}$

Note 26: MSW LFSR Seed <15:0> is the MS word of the LFSR seed.

Note 27: This register along with register LSSEDR form a register pair and should be read/ written with two consecutive read/write accesses.

TABLE 28. TDI_SM LFSR LSB Result Register (LSRESR) ($0A) (Notes 28, 29)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15:0	RW	LSW LFSR Result	0	0000h	$\overline{\text{SYS\_RST}}$

Note 28: LSW LFSR Result<15:0> is the LS word of the LFSR result.

Note 29: This register along with register MSRESR form a register pair and should be read/written with two consecutive read/write accesses.

TABLE 29. TDI_SM LFSR MSB Result Register (MSRESR) ($0B) (Notes 30, 31)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15:0	RW	MSW LFSR Result	0	0000h	$\overline{\text{SYS\_RST}}$

Note 30: MSW LFSR Result<15:0> is the MS word of the LFSR result.

Note 31: This register along with register LSRESR form a register pair and should be read/ written with two consecutive read/write accesses.

TABLE 30. Index Register (INDEXR) ($0C) (Notes 32, 33, 34)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15:0	RW	Index	0	0000h	$\overline{\text{SYS\_RST}}$

Note 32: Index<15:0> sets the individual address memory pointer.

Note 33: Address memory pointer must be on a long word boundary.

Note 34: Writing to this register sets the TDO_SM, TDI_SM, Expected and Mask pointers. These pointers will automatically increment with each long word read from the TDI_SM space and each long word write to the other TDO_SM, Expected and Mask spaces.

TABLE 31. Vector Index Register (VINDEXR) ($11) (Notes 35, 36)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15:0	RW	Vector Index	0	0000h	$\overline{\text{SYS\_RST}}$

Note 35: Vector Index<15:0> sets the Vector address memory pointer.

Note 36: Address memory pointer must be on a long word boundary.

TABLE 32. Header/Trailer Index Register (HTINDEXR) ($13) (Notes 37, 38)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15:0	RW	Header/Trailer Index	0	0000h	$\overline{\text{SYS\_RST}}$

Note 37: Header/Trailer Index<15:0> sets the Header/Trailer address memory pointer.

Note 38: Address memory pointer must be on a long word boundary.

TABLE 33. Macro Index Register (MINDEXR) ($15) (Notes 39, 40)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15:0	RW	Macro Index	0	0000h	$\overline{\text{SYS\_RST}}$

Note 39: Macro Index<15:0> sets the Macro address memory pointer.

Note 40: Address memory pointer must be on a long word boundary.

TABLE 34. Sequencer Index Register (SINDEXR) ($17) (Notes 41, 42)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15:0	RW	Sequencer Index	0	0000h	$\overline{\text{SYS\_RST}}$

Note 41: Sequencer Index<15:0> sets the Sequencer address memory pointer.

Note 42: Address memory pointer must be on a long word boundary.

TABLE 35. ScanBridge Support Index Register (BSINDEXR) ($19) (Notes 43, 44)

Bit(s)	Type	Field	Address Offset	Reset Value	Reset Source
15:0	RW	ScanBridge Index	0	0000h	$\overline{\text{SYS\_RST}}$

Note 43: ScanBridge Index<15:0> sets the ScanBridge Support address memory pointer.

Note 44: Address memory pointer must be on a long word boundary.

Testability Details - IEEE 1149.1 Support

An 8 instruction Tap Controller will be used to accomplish the IEEE 1149.1 support design.

TABLE 36. Supported IEEE 1149.1 Instruction Set

Instruction Mnemonic	Binary Instruction Code	Description
EXTEST	000	Allows off-chip circuitry and interconnect to be tested.
SAMPLE/PRELOAD	001	Allows snapshot of normal operation. Also allows data to be loaded on parallel output boundary scan registers.
BYPASS	111	Places device in bypass mode so that there is single shift register stage between TDI and TDO.
IDCODE	010	Allows scanning of the device identification register.
HIGHZ	011	Tristates all output drivers with the exception of TDO.
CLAMP	100	Allows the state of the signals driven from component pins to be determined from the boundary-scan register while the BYPASS register is selected as the serial path between TDI and TDO.
RUNBIST	110	Enables on chip BIST logic to perform memory BIST.
SCANTEST	101	Allows the assertion of internal test_mode signal to prevent the asynchronous resets from inadvertantly resetting the flip-flops during internal scan.

TABLE 37. IDCODE Register Description

Version	Part Number	Manufacturer Identity	Start Bit
"0000"	"1111 1100 0001 0111"	"000 0000 1111"	"1"

TABLE 38. Boundary Scan Register Definition

BSR Bit#	Signal Name	BSR Bit#	Signal Name	BSR Bit#	Signal Name	BSR Bit#	Signal Name
0	SCK	10	$\overline{\text{DTACK}}$	20	DATA[7]	30	TCK_SM
1	$\overline{\text{RST}}$	11	INT	21	DATA[6]	31	$\overline{\text{TRST0\_SM}}$
2	R/$\overline{\text{W}}$	12	DATA[15]	22	DATA[5]	32	TDI_SM
3	$\overline{\text{STB}}$	13	DATA[14]	23	DATA[4]	33	$\overline{\text{OE}}$
4	$\overline{\text{CE}}$	14	DATA[13]	24	DATA[3]	34	TRIST
5	ADDRESS[4]	15	DATA[12]	25	DATA[2]		
6	ADDRESS[3]	16	DATA[11]	26	DATA[1]		
7	ADDRESS[2]	17	DATA[10]	27	DATA[0]		
8	ADDRESS[1]	18	DATA[9]	28	TDO_SM		
9	ADDRESS[0]	19	DATA[8]	29	TMS_SM		

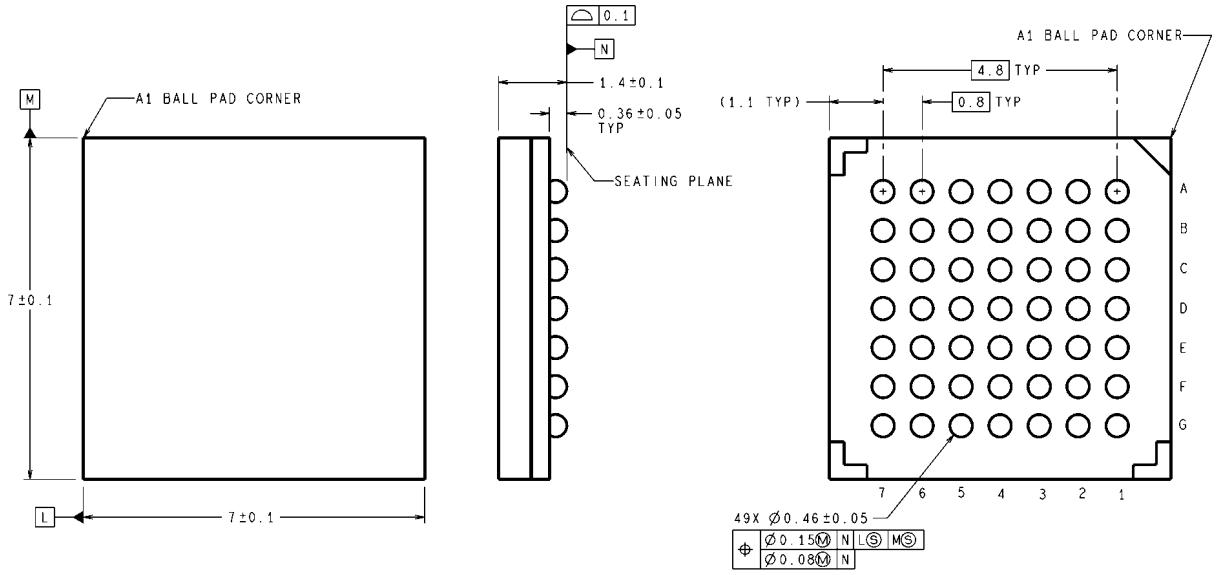
BIST SUPPORT

The memory BIST can be initiated through JTAG interface using the RUNBIST instruction or by setting the Onboard Memory BIST bit in the Start register. When the memory BIST is initiated through the JTAG interface the result of pass/fail will be set in the Memory BIST Result bit in the Status register and also in the BIST status register that can be accessed through the JTAG interface. The BIST status register is a one bit register and is connected in the serial path of TDO and TDI when RUNBIST instruction is scanned into the instruction register. Once the BIST is done the contents of the BIST status register can be scanned out to determine whether the memory BIST passed or failed. If the memory BIST is initiated through the Onboard Memory BIST bit in the Start register the result of pass/fail will be set only in the BIST Result bit in the status register. The memory BIST will initialize the memory to zero.

SCAN METHODOLOGY

The STA101 supports internal scan through the shared ports SCAN_EN, SCAN_IN, SCAN_OUT. Before initiating an internal scan test the user should scan in SCANTEST instruction through the JTAG interface so that an internal test_mode signal can be asserted. This test_mode signal is used to prevent the reset from inadvertantly resetting the flip-flops during internal scan. The test vectors to verify the scan chain are generated by the Sunrise test tool. The target for the stuck-at fault coverage is 97% and the achieved fault coverage is about 99%.

Physical Dimensions inches (millimeters) unless otherwise noted



DIMENSIONS ARE IN MILLIMETERS

SLC49A (Rev B)

49-Pin BGA
NS Package Number SLC49A
Ordering Code SCANSTA101SM
(Tape and Reel Ordering Code SCANSTA101SMX)

Notes

National Semiconductor Americas Customer Support Center
Email: new.feedback@nsc.com
Tel: 1-800-272-9959

National Semiconductor Europe Customer Support Center
Fax: +49 (0) 180-530-85-86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +49 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

National Semiconductor Asia Pacific Customer Support Center
Email: ap.support@nsc.com

National Semiconductor Japan Customer Support Center
Fax: 81-3-5639-7507
Email: jpn.feedback@nsc.com
Tel: 81-3-5639-7560