

Boundary Scan User's Guide

Release 02.2025



TRACE32 Online Help	
TRACE32 Directory	
TRACE32 Index	
TRACE32 Documents	Þ
Boundary Scan	þ
Boundary Scan User's Guide	1
Introduction	4
Intended Audience	4
How This Manual is Organized	5
Related Documents	5
List of Abbreviations	5
What to know about Boundary Scan	6
Configuration of the Boundary Scan Chain	8
Configure Boundary Scan Engine	8
Interaction with Debug Function	8
The Parking State	9
Loading the BSDL Files	10
Initialization of the Boundary Scan Chain	11
Check the Configuration	12
General Operation	13
Basic Mode of Operation	13
Preparation of the Boundary Scan Chain	13
Modification of Instruction Registers	13
Modification of a Data Register	14
Modification of the Boundary Scan Register	15
The BSDL.SET BSR Command	15
The BSDL.SET PORT Command	16
Execution of the Boundary Scan Commands	17
Working with the GUI	18
The BSDL.state Window	18
Configure Tab	18
Check Tab	19
Run Tab	19
The BSDL.SET Window	21
Sample View	22
Set Write View	22

Set Read View	24
File info	25
Interactive Board Test	26
Configure Run Mode	27
Set and Run Mode	27
Two Step DR Mode	28
Execute Level and Connection Tests	30
Level Tests	30
Connection Tests	31
Automated Board Test	33
Prepare Boundary Scan Chain	34
Run Tests	35
Full Example	36
Test of Non-Boundary Scan Devices	38
Special Tests	42
Boundary Scan Oscilloscope	42
Other Instructions and Data Registers	43
Tips and Tricks	45

Version 13-Feb-2025

Introduction

TRACE32 boundary scan function supports:

- Interactive board connection test
- Automated board connection test
- FLASH programming (See "FLASH Programming via Boundary Scan" in Onchip/NOR FLASH Programming User's Guide, page 88 (norflash.pdf).

Additionally, all boundary scan instructions and data registers, which are described in the BSDL file can be accessed.

Intended Audience

PCB developers who want to:

- Check interconnects and signals for initial operation of their board
- Test board mounting for developer and engineering boards

Software developers who want to:

- Measure port states
- Drive signals on printed circuit boards (PCBs)
- Program and read out external flash memories

IC manufacturers who want to:

- Run built-in self tests of their ICs
- Enable special test features, which can be controlled via boundary scan

How This Manual is Organized

- "What to know about Boundary Scan": A short introduction to the boundary scan.
- "Configuration of the Boundary Scan Chain": Describes how to set up a boundary scan chain and do basic checks.
- "General Operation": Describes boundary scan operation and working with the BSDL GUI.
- "Interactive Board Test": Describes how to do interactive board testing.
- "Automated Board Test": Describes how to set up a test for a printed circuit board (PCB) script and run the test.
- "Special Tests": Describes more test and debug features.

Related Documents

For information about NOR FLASH programming, please refer to:

• "Onchip/NOR FLASH Programming User's Guide" (norflash.pdf)

For information about eMMC programming, please refer to:

"eMMC FLASH Programming User's Guide" (emmcflash.pdf)

For information about serial FLASH programming, please refer to:

• "Serial FLASH Programming User's Guide" (serialflash.pdf)

For information about the boundary scan commands (BSDL), please refer to:

BSDL in "General Commands Reference Guide B" (general_ref_b.pdf)

List of Abbreviations

BSDL	Boundary Scan Description Language			
DR	<u>D</u> ata <u>R</u> egister			
IC	Integrated <u>C</u> ircuit			
IR	Instruction <u>R</u> egister			
РСВ	Printed <u>C</u> ircuit <u>B</u> oard			
ТАР	<u>T</u> est <u>A</u> ccess <u>P</u> ort			

тск	<u>T</u> est <u>C</u> loc <u>k</u> Input
TDI	<u>T</u> est <u>D</u> ata <u>I</u> nput
TDO	<u>T</u> est <u>D</u> ata <u>O</u> utput
TMS	<u>T</u> est <u>M</u> ode <u>S</u> elect
TRST_N	<u>T</u> est <u>R</u> e <u>s</u> et Input

What to know about Boundary Scan

Boundary scan is a method for testing interconnects on PCBs and internal IC sub-blocks. It is defined in the IEEE 1149.1 standard.

For boundary scan tests, additional logic is added to the device. Boundary scan cells are placed between the core logic and the ports.



In normal mode, these cells are transparent and the core is connected to that ports. In boundary scan mode, the core is isolated from the ports and the port signals are controlled by the JTAG interface.

The boundary scan cells are connected to a serial shift register, which is referred to as the boundary scan register. This register can be used to read and write port states.

Following boundary scan instructions are defined in the IEEE standard:

- BYPASS (mandatory): TDI is connected to TDO via a single shift register.
- SAMPLE (mandatory): Takes a snapshot of the normal operation of the IC.
- PRELOAD (mandatory): Loads data to the boundary scan register.
- EXTEST (mandatory): Apply preloaded data of the boundary scan register to the ports.
- INTEST (optional): Apply preloaded data of the boundary scan register to the core logic.
- RUNBIST (optional): Executes a self-contained self-test of the IC.
- CLAMP (optional): Apply preloaded data of the boundary scan register to the ports and selects the bypass register as the serial path between TDI and TDO.
- IDCODE (optional): Reads the device identification register.
- USERCODE (optional): Reads and writes a user programmable identification register.
- HIGHZ (optional): Places the IC in an inactive drive state (e.g. all ports are set to high impedance state).

In addition, IC specific instructions may be defined.

The structure of the boundary scan chain and the instruction set are described with the Boundary Scan Description Language (BSDL). BSDL is a subset of the Very High-level Design Language (VHDL). The BSDL files are provided by the IC manufacturer.

The steps below provide an overview of configuration process and are described in detail in the following sections.

- 1. Configure and initialize Boundary scan engine. (See Configure Boundary Scan Engine)
- 2. Load the BSDL files in the correct order. (See "Loading the BSDL Files".)
- 3. Initialize the boundary scan chain. (See "Initialization of the Boundary Scan Chain".)
- 4. Check the configuration of the boundary scan chain. (See "Check the Configuration".)

Configure Boundary Scan Engine

To start from a defined state use command **BSDL.RESet** first. It will initialize the boundary scan engine to:

- Empty boundary scan chain (i.e. all loaded BSDL files will be removed)
- Removes any previously defined FLASH configuration
- Sets parking state to Run-Test/Idle
- Disables set-and-run mode
- Disables two-step-DR mode
- Disables result checking
- Sets the current state of the boundary scan chain to unknown (i.e. a BSDL.SOFTRESET or BSDL.HARDRESET command must executed first to start boundary scan tests)

Interaction with Debug Function

If only boundary scan functions are used, the debug functions should be turned off with:

; disable the debugger SYStem.Down



If boundary scan and debugging should be used at the same time, the boundary scan configuration and initialization should be done prior debug configuration:

```
; disable the debugger
SYStem.Down
;Initialize the boundary scan engine
BSDL.RESet
;Load the BSDL file(s)
BSDL.FILE file1.bsdl
;Reset the boundary scan chain
BSDL.SOFTRESET
; now configure the debugger
...
```

It is also required to set the boundary scan instruction register each time when a boundary scan data register is accessed (i.e. use always **BSDL.RUN** and not **BSDL.RUN DR**).

The Parking State

Each boundary scan operation starts and stops from the parking state. The default parking state is the Run-Test/Idle state. It can be modified with the **BSDL.ParkState** command.

If the boundary scan park state is different from the debug park state, unintended side effects may occur. To avoid this, it is recommended to set the parking state of the boundary scan engine to the parking state of the debug engine:

```
; initialize the boundary scan engine
BSDL.RESet
; Set the parking state to Select-DR-Scan (e.g. for Arm or MIPS)
BSDL.ParkState Select-DR-Scan
```

If it is not possible to use the same parking state and only boundary scan functions are used, the JTAG interface should be locked:

```
; disable the debugger and lock the JTAG interface
SYStem.Down
JTAG.LOCK
;Initialize the boundary scan engine
BSDL.RESet
;Load the BSDL file(s)
BSDL.FILE file1.bsdl
;Reset the boundary scan chain
BSDL.SOFTRESET
; do boundary scan functions
...
;unlock the JTAG interface
JTAG.UNLOCK
```

The boundary scan chain of a PCB is configured by loading the required BSDL files in the correct order. The BSDL file for the IC connected to the TDO port of the PCB must be loaded first (IC1). The IC connected to the TDI port must be loaded as the last file.



Use the following command to open the configuration window for boundary scan tests:

BSDL.state

Opens the **BSDL.state** window.

B::BSDL.state		C	
Configure Chec	k Run		
	IOVEUP JtagClo	ock JTAG.I	LOCK
			Ked
No. Entity	Instruction	DR Name	DR Size
1 ispPAC_CLK5610AV	X BYPASS*	BYPASS*	1 🔺
2 ispPAC_CLK5610AV	_X+ BYPASS*	BYPASS*	1
3 ISPPAC_CLK5610AV	_X BYPASS*	BYPASS*	1 -
4 LCMX01200C_XXF25	6 BYPASS*	BYPASS*	1
			-

The **FILE** button adds a boundary scan file to the current position, the **UNLOAD** button removes the currently selected file.

Within the **BSDL.state** window, the currently selected file can be positioned with the **MOVEUP** and **MOVEDOWN** buttons. The list shows the current configuration of the boundary scan chain, i.e. the list entry no.1 is IC1 (connected to TDO of the PCB), the last list entry is connected to the TDI of the PCB.

Alternatively, the BSDL files can be loaded with the command BSDL.FILE.

BSDL.FILE ispCLOCK5610Av_isc.bsm	; Load BSDL file for IC1
BSDL.FILE ispCLOCK5610Av_isc.bsm	; Load BSDL file for IC2
BSDL.FILE ispCLOCK5610Av_isc.bsm	; Load BSDL file for IC3
BSDL.FILE LCMXCO1200C_ftBGA256.bsdl	; Load BSDL file for IC4

After loading the BSDL files, the boundary scan chain should be initialized.

BSDL.SOFTRESET	;	do a	а	sequential	TAP	reset	(through	TAP
	;	stat	te	e Test-Logio	c-Re	set)		

Some ICs may require a reset via TRST_N (e.g. to latch the device into boundary scan mode).

BSDL.HARDRESET	;	do	an	asynchronous	TAP	reset	via	TRST_	N
----------------	---	----	----	--------------	-----	-------	-----	-------	---

Both commands will reset the TAP controllers for all ICs in the boundary scan chain and halt in *Select-DR-SCAN* state. Usually only one of these commands is required, only if an IC in the boundary scan chain requires a TRST_N reset and at least one IC in the boundary scan chain has no TRST_N port, then apply both commands in the following order:

BSDL.HARDRESET

BSDL.SOFTRESET

After the configuration of the boundary scan chain, the BYPASS mode and the IDCODE of all ICs in the chain should be checked. This means that the debugger verifies if JTAG works correctly and if the BSDL files match the selected ICs.

On tab **Check** of the **BSDL.state** window these checks can be done with the buttons **BYPASSall** and **IDCODEall**. The check results are shown next to the buttons.



Alternatively, the checks can be performed with the following commands:

BSDL.BYPASSall	;	Check	BYPA	SS	mode	e fo	or	al	l ICs
BSDL.IDCODEall	;	Check	the	IDC	CODE	of	al	1 :	ICs

If a check fails, an error will be reported.

For PRACTICE scripts, the functions **BSDL.CHECK.BYPASS()** and **BSDL.CHECK.IDCODE()** can be used for checking *BYPASS* and *IDCODE* of the boundary scan chain.

```
if !bsdl.check.bypass()
    PRINT %ERROR "Bypass test failed"
else
(
    if !bsdl.check.idcode()
        PRINT %ERROR "ID code test failed"
    else
    (
        ...
    ; further commands
```

Basic Mode of Operation

A boundary scan step consists of two phases:

- 1. Preparation of the boundary scan chain with the **BSDL.SET** command
- 2. Execution of the boundary scan instruction and data settings with the **BSDL.RUN** command

During the preparation phase, instruction and data register settings are modified, but not shifted to the boundary scan chain. All modifications are stored in shadow registers on the host side. For each data register in the BSDL file there is a set of shadow registers, which will store the status of the boundary scan chain until the end of the boundary scan test.

With the **BSDL.RUN** command, the current status of the boundary scan chain is shifted to the hardware registers of the ICs on the PCB.

Preparation of the Boundary Scan Chain

The general command for the preparation of the boundary scan chain is:

BSDL.SET <chip_number> <set_selection>

Modify settings *<set_selection>* for IC *<chip_number>*

An arbitrary number of **BSDL.SET** commands can be applied before execution. All modifications are incremental, i.e. a previously modified register bit is only overwritten, if a **BSDL.SET** command addresses the same bit.

The boundary scan chain is prepared by modifying the instruction registers, the data register and the boundary scan register, as described in the following sections.

Modification of Instruction Registers

An instruction register is modified with:

BSDL.SET <chip_number> IR <instrname>

Set instruction *<instrname>* for IC *<chip_number>*

Example:

BSDL.SET 3. IR BYPASS

; Set instruction BYPASS for IC3

Instructions can be only set, if they are defined in BSDL file of the IC *<chip_number>*. If an instruction register is modified, the appropriate fields in the **BSDL.state** window will change their color to red.

B	::BSDL.state		c	
[Configure Check	Run		
	FILE MOVE	UP JtagClo DOWN 20.0N	ock JTAG.I 1Hz - Oc	LOCK ked
No.	Entity	Instruction	DR Name	DR Size
2 3	ispPAC_CLK5610AV_X ispPAC_CLK5610AV_X ispPAC_CLK5610AV_X	IDCODE IDCODE BYPASS*	DEVICE_ID DEVICE_ID BYPASS*	³² ³² 1
4	LCMX01200C_XXF256	IDCODE	DEVICE_ID	32 •
1				н. 4

Modification of a Data Register

The general command for modifying a data register is:

BSDL.SET <chip_number> DR <bitslice> <setvalue> Set <bitslice> of current data register to value <setvalue> for IC <chip_number>

The *<bitslice>* can be a single bit, a range (i.--k.) or the whole data register (*). *<setvalue>* can be **ZERO**, **ONE** or a hexadecimal number for write data register. With **ExpectH**, **ExpectL** and **ExpectX** the expected value for the read data register can be set.

To check the values of the read data register, set the result check to ON:

```
BSDL.CHECK ON ; Enables result checking
```

Examples:

BSDL.SET 4. DR * ZERO	; Set all bits of the current data ; register of IC4 to zero
BSDL.SET 4. DR 410. ONE	; Set bits 4-10 of the current data ; register of IC4 to one
BSDL.SET 4. DR 1632. 0xa5a5	; Set bits 16-32 of the current data ; register of IC4 to 0xa5a5

The boundary scan register can only be modified if one of the following instructions is set: *SAMPLE*, *PRELOAD*, *EXTEST*, *INTEST*).

There are two ways to modify the boundary scan register:

- 1. The **BSDL.SET BSR** command.
- 2. The **BSDL.SET PORT** command.

The BSDL.SET BSR Command

With the **BSDL.SET BSR** command bit slices of the boundary scan register can be modified:

BSDL.SET <chip_number> BSR <bitslice> <setvalue></setvalue></bitslice></chip_number>	Set <i><bitslice></bitslice></i> of the boundary scan register to value <i><setvalue></setvalue></i> for IC <i><chip_number></chip_number></i>

It has the following <setvalues>:

- ZERO, ONE: sets < bitslice> to zero or one
- SAFE: sets < bitslice> to SAFE state acc. BSDL file
- SAMPLE: sets < bitslice> to previously sampled data
- DISable, ENable: disables/enables output drivers for all ports in
 bitslice>
- Drive0, Drive1: drives 0/1 for all ports in

 bitslice>
- ExpectX, ExpectH, ExpectL: sets expected read data of < bitslice> to ignore (X)/high (H)/low (L)
- Hexadecimal number: sets < bitslice> to this value

Besides these additional <setvalues> all other <setvalues> of the BSDL.SET DR command can be used.

BSDL.SET 4. BSR * ZERO	; Set all bits of the boundary scan ; register of IC4 to zero
BSDL.SET 4. BSR 410. SAFE	; Set bits 4-10 of the boundary scan ; register of IC4 to SAFE state
BSDL.SET 4. BSR 1632. SAMPLE	; Set bits 16-32 of the boundary scan ; register of IC4 to previously ; sampled data

Use the BSDL.SET PORT command for the direct modification of ports:

BSDL.SET <chip_number> **PORT** <port_name> <port_value>

Set port <port_name> to value <port_value> for IC <chip_number>

The <port_name> must be a valid port name from the BSDL file of the IC <chip_number>.

BSDL.SET 4. PORT PL7A 1	; Set port PL7A of IC4 to drive 1
BSDL.SET 3. PORT PS1 H	; Set port PS1 of IC3 to read high level

After completing the preparations of the boundary scan chain, you can apply the modified settings to the hardware registers of the boundary scan chain.

BSDL.RUN	;	Execute	an	IR-Scan	and	then	а
	;	DR-SCAN	ope	eration			

To apply only the instruction register settings to the boundary scan chain, use this command:

BSDL.RUN IR	;	Execute	an	IR-SCAN	operation
-------------	---	---------	----	---------	-----------

To apply only the data register settings to the boundary scan chain, use this command:

BSDL.RUN DR ; Execute a DR-SCAN operation

If the result check is set to **ON**, the bits of the read data registers of all ICs in the boundary scan chain will be compared to the expected values. If the result mismatches, an error will be reported at the message line. For more details about the error, open the **AREA.view** window

The BSDL.state Window

The **BSDL.state** window consists of two parts: the dialog area with 3 tabs **Configure**, **Check**, **Run** and the list with the loaded boundary scan structures.

Configure Tab

Use the configure tab to configure the boundary scan:

- Load BSDL files
- Remove boundary scan structures from the boundary scan chain
- Arrange boundary scan structures
- Set JTAG clock (TCK frequency)
- Lock JTAG interface



In the list area the loaded boundary scan structures for the ICs in the boundary scan chain are shown. The columns give information about:

- No.: Position of the IC in the boundary scan chain (TDI -> ICn -> ... -> IC1 -> TDO)
- Entity: Name of the IC entity (from the BSDL file)
- **Instruction**: Current instruction of the IC
- **DR Name**: Name of the current data register of the IC (depending on the current instruction)
- **DR Size**: Size of the current data register of the IC (depending on the current instruction)

On the **Check** tab, you can start the checks of the boundary scan chain by clicking the buttons **BYPASSall**, **IDCODEall**, and **SAMPLEall**.

The check results are displayed next to the buttons as shown in the screenshot below.



Max. TCK for the boundary scan chain (from the BSDL files) and compare result with current TCK

The **BYPASSall** check performs a *BYPASS* test of the boundary scan chain.

The **IDCODEall** check performs an *IDCODE* test for all ICs in the boundary scan chain.

The **SAMPLEall** check takes a *SAMPLE* from all ICs in the boundary scan chain.

The check results can be:

- No result: test was not executed (BYPASSall, IDCODEall, SAMPLEall)
- Test PASS: test was executed and passed (BYPASSall, IDCODEall)
- Test FAIL: test was executed and failed (BYPASSall, IDCODEall)
- Test done: test was executed (SAMPLEall)

Run Tab

On the **Run** tab, you can execute the boundary scan settings by clicking the buttons **RUN IR**, **RUN DR**, and **RUN**. You can release **SOFTRESET** and **HARDRESET** for the boundary scan chain and set execution options:

- CHECK: enable/disable read result checking
- SetAndRun: enable/disable immediate takeover of data register modifications
- **TwoStepDR**: enable/disable two *DR-SCAN* cycles for **RUN/RUN DR** command

In addition, you can set instructions by right-clicking the desired row and selecting an entry from the context menu (see screenshot below).

B	B::BSDL.state			- 0	×		
	Configure Check RUN IR RUN RUN DR CHECK	Run	un SOF	TRESET			Context menu for quick modification of the ICs instruction Only BYPASS.
							SAMPLE and
No.	Entity	Instruction	DR Name	DR Si	ze		IDCODE possible
1	ispPAC_CLK5610AV_X ispPAC_CLK5610AV_X	BYPASS	BYPASS		1		other settings must be
3	ispPAC_CLK5610AV_X	SAMPLE	BOUNDARY		Instru	uction	
4	LCMX01200C_XXF256	EXTEST	BOUNDARY	4	ВҮРА	SS	done in the BSDL.SET
1					SAM	PLE	window
				_	IDCO	DE	

With the context menu, you can set the 3 "read-only" instructions (*BYPASS, SAMPLE, IDCODE*) for the selected list entry.

The **BSDL.SET** window for an IC is either opened with a double click on the IC's entry in the **BSDL.state** window or with the command:

```
BSDL.SET 3. ; Opens the configuration and result ; window for IC 3
```

In the **BSDL.SET** window all instructions defined in the BSDL file can be set, options can be selected, data register bits can be set and the data register results can be viewed.

The content of the data area depends on the selected instruction, the Filter data settings and the DR mode.

The data area provides the following views:

- 1. Sample view
- 2. Set Write view
- 3. Set Read view
- 4. File Info

Instruction				
selection list	BSDL file name	View 1-3	View 4	
		/		
B::BSDL.SET 3.				
BSDL Configuration	- ispPAC_CLK5610AV_XXT48]	🔰 File Info	
Instructions USERCODE ISC_ADDRESS_IV ISC_ADDRESS_SI ISC_DATA_SHIFT	Data format ibin hex	DR mode Init DR Sample ExpectX Set Write Set Read ExpectL		Status line (data register
Data register bit Expect 0 H 1 H 2 H 3 H 4 H 5 H	ISC_ADDRESS (expect, /alue L L L (PASS) L H (PASS) L H (PASS) L L (PASS)	single step DR mode, s	set & run mode disabled)	name, mode, run options)
6 7 8 9 H				–Data area

In the **Sample** view the results of the last data register read operation is shown. With **Spotlight** enabled, register bit changes are highlighted.

BSDL Configuration - ispPAC_CLK5610AV_XXT48 ▲ File Info Instructions Data format DR mode Filter data BYPASS	B::BSD	L.SET 3.]
Instructions Data format DR mode Filter data BYPASS in in in in in SAMPLE in in in in in in PRELOAD in in in in in in in PRELOAD in in in in in in in in PRELOAD in Pata register: BOUNDARY (sample) in	BSDL Cor	figuration - ispPAC	CLK561	LOAV_XX	T48				🔰 File Info]
Data register: BOUNDARY (sample) num port pin pintype function Reg. Enable 1 LOCK 34 OUT OUTPUT3 1 en 2 SGATE 40 IN INPUT 0 en 3 RESET 41 IN INPUT 0 en 4 GOE 42 IN INPUT 0 en 5 PS1 43 IN INPUT 0 en 6 PS0 444 IN INPUT 1 en 9 BANK_0A 3 OUT OUTPUT3 0 en 11 BANK_0B 2 OUT OUTPUT3 0 en 17 BANK_1A 7 OUT OUTPUT3 0 en 13 BANK_2B 10 OUT OUTPUT3 0 en 25 BANK_2B 10 OUT OUTPUT3 <td< td=""><td>BYPAS SAMPL PRELO, EXTES</td><td>tions S AD T</td><td>─ Data ⓒ bir ◉ he</td><td>format - 1 x</td><td>DR moo</td><td>de — Filt ple II Vrite I C Read I B</td><td>er data — nput 🔽 hutput 🕅 idi 🔽</td><td>Observe Intern Spotlight</td><td></td><td></td></td<>	BYPAS SAMPL PRELO, EXTES	tions S AD T	─ Data ⓒ bir ◉ he	format - 1 x	DR moo	de — Filt ple II Vrite I C Read I B	er data — nput 🔽 hutput 🕅 idi 🔽	Observe Intern Spotlight		
num port pin pintype function Reg. Enable 1 LOCK 34 OUT OUTPUT3 1 en ^ 2 SGATE 40 IN INPUT 0 en ^ 3 RESET 41 IN INPUT 0 en ^ 4 GOE 42 IN INPUT 0 en ^ 5 PS1 43 IN INPUT 0 en	Data r	egister: BOUND	DARY (s	ample)						
1 LOCK 34 OUT OUTPUT3 1 en 2 SGATE 40 IN INPUT 0 en 3 RESET 41 IN INPUT 0 en 4 GOE 42 IN INPUT 0 en 5 PS1 43 IN INPUT 0 en 6 PS0 44 IN INPUT 0 en 7 PLL_BYPASS 47 IN INPUT 0 en 9 BANK_0A 3 OUT OUTPUT3 0 en 11 BANK_1B 6 OUT OUTPUT3 0 en 17 BANK_1B 6 OUT OUTPUT3 0 en 19 BANK_2A 11 OUT OUTPUT3 0 en 27 BANK_2B 10 OUT OUTPUT3 0 en 28 FBKA_P 15 IN INPUT 0 en 34 OEX 21	num	port	10031020160	pin	pintype	function	Reg.	Enable		1
2 SGALE 40 IN INPUT 0 en 3 RESET 41 IN INPUT 0 en 4 GOE 42 IN INPUT 0 en 5 PS1 43 IN INPUT 0 en 6 PS0 44 IN INPUT 1 en 7 PLL_BYPASS 47 IN INPUT 1 en 9 BANK_0A 3 OUT OUTPUT3 0 en 11 BANK_1B 6 OUT OUTPUT3 0 en 19 BANK_1A 7 OUT OUTPUT3 0 en 25 BANK_2A 11 OUT OUTPUT3 0 en 27 BANK_2B 10 OUT OUTPUT3 0 en 28 FBKA_P 15 IN INPUT 0 en 34 OEX 21 IN INPUT 0 en 35 OEY 22 <	1	LOCK		34	OUT	OUTPUT3	1	en		
3 RESET 41 IN INPUT 0 en 4 GOE 42 IN INPUT 0 en 5 PS1 43 IN INPUT 0 en 6 PS0 44 IN INPUT 1 en 7 PLL_BYPASS 47 IN INPUT 1 en 9 BANK_0A 3 OUT OUTPUT3 0 en 11 BANK_0B 2 OUT OUTPUT3 0 en 19 BANK_1A 7 OUT OUTPUT3 0 en 25 BANK_2A 11 OUT OUTPUT3 0 en 25 BANK_2B 10 OUT OUTPUT3 0 en 26 FBKA_P 15 IN INPUT 0 en 34 OEX 21 IN INPUT 0 en 35 OEY 22 IN INPUT 0 en 37 BANK_3A 27	1 5	SGATE		40	IN	INPUT	0	en		
7 GOE 72 IN INPUT 0 en 5 PS1 43 IN INPUT 1 en 6 PS0 44 IN INPUT 0 en 7 PLL_BYPASS 47 IN INPUT 1 en 9 BANK_0A 3 OUT OUTPUT3 0 en 11 BANK_0B 2 OUT OUTPUT3 0 en 11 17 BANK_1B 6 OUT OUTPUT3 0 en 25 BANK_2A 11 OUT OUTPUT3 0 en 26 27 BANK_2B 10 OUT OUTPUT3 0 en 28 FBKA_P 15 IN INPUT 0 en 34 OEX 21 IN INPUT 0 en 35 OEY 22 IN INPUT 0 en 39 BANK_3A 27 OUT OUTPUT3 0 en 37		COE		41		TNDUT	N N	en		E
6 PSD 44 IN INPUT 1 en 7 PLL_BYPASS 47 IN INPUT 1 en 9 BANK_0A 3 OUT OUTPUT3 0 en 11 BANK_0B 2 OUT OUTPUT3 0 en 17 BANK_1A 7 OUT OUTPUT3 0 en 19 BANK_1B 6 OUT OUTPUT3 0 en 25 BANK_2A 11 OUT OUTPUT3 0 en 27 BANK_2B 10 OUT OUTPUT3 0 en 28 FBKA_P 15 IN INPUT 0 en 31 REFA_P 18 IN INPUT 0 en 34 OEX 21 IN INPUT 0 en 37 BANK_3A 27 OUTPUT3 0 en 1 39 BANK_3B 26 OUT OUTPUT3 0 en 45 BANK_4A	1 2	DS1		43		TNDUT	1	en		
7 PLL_BYPASS 17 11 INPUT 1 en 9 BANK_0A 3 OUT OUTPUT3 0 en 11 BANK_0B 2 OUT OUTPUT3 0 en 17 BANK_1A 7 OUT OUTPUT3 0 en 19 BANK_1A 7 OUT OUTPUT3 0 en 25 BANK_1B 6 OUT OUTPUT3 0 en 27 BANK_2B 10 OUT OUTPUT3 0 en 28 FBKA_P 15 IN INPUT 0 en 34 OEX 21 IN INPUT 0 en 35 OEY 22 IN INPUT 0 en 39 BANK_3A 27 OUTPUT3 0 en	i é	PS0		44	TN	TNPUT	0	en		L
9 BAKK_0A 3 OUT OUTPUT3 0 en 11 BANK_0B 2 OUT OUTPUT3 0 en 17 BANK_1A 7 OUT OUTPUT3 0 en 19 BANK_1B 6 OUT OUTPUT3 0 en 25 BANK_2A 11 OUT OUTPUT3 0 en 27 BANK_2B 10 OUT OUTPUT3 0 en 28 FBKA_P 15 IN INPUT 0 en 31 REFA_P 18 IN INPUT 0 en 34 OEY 22 IN INPUT 0 en 35 OEY 22 IN INPUT 0 en 39 BANK_3A 27 OUTPUT3 0 en 45 BANK_4A 31 OUT OUTPUT3 0 en	ĬŽ	PLL RYPASS		47	TN	TNPUT	ĭ	en		
11 BANK_0B 2 OUT OUTPUT3 0 en 17 BANK_1A 7 OUT OUTPUT3 0 en 19 BANK_1B 6 OUT OUTPUT3 0 en 25 BANK_2A 11 OUT OUTPUT3 0 en 27 BANK_2B 10 OUT OUTPUT3 0 en 28 FBKA_P 15 IN INPUT 0 en 31 REFA_P 18 IN INPUT 0 en 34 OEX 21 IN INPUT 0 en 35 OEY 22 IN INPUT 0 en 37 BANK_3A 27 OUT OUTPUT3 0 en 39 BANK_3B 26 OUT OUTPUT3 0 en 45 BANK_4A 31 OUT OUTPUT3 0 en	9	BANK OA		3	OUT	OUTPUT3	ō	en		
17 BANK_1A 7 OUT OUTPUT3 0 en 1 19 BANK_1B 6 OUT OUTPUT3 0 en 1 1 25 BANK_2A 11 OUT OUTPUT3 0 en 1	11	BANK_0B		2	OUT	OUTPUT3	0	en		
19 BANK_1B 6 OUT OUTPUT3 0 en 25 BANK_2A 11 OUT OUTPUT3 0 en 27 BANK_2B 10 OUT OUTPUT3 0 en 28 FBKA_P 15 IN INPUT 0 en 31 REFA_P 18 IN INPUT 0 en 34 OEX 21 IN INPUT 0 en 35 OEY 22 IN INPUT 0 en 37 BANK_3A 27 OUT OUTPUT3 0 en 39 BANK_3B 26 OUT OUTPUT3 0 en 45 BANK_4A 31 OUT OUTPUT3 0 en	17	BANK_1A		7	OUT	OUTPUT3	0	en	=	
25 BANK_2A 11 OUT OUTPUT3 O en 27 BANK_2B 10 OUT OUTPUT3 O en 28 FBKA_P 15 IN INPUT O en 31 REFA_P 18 IN INPUT O en 34 OEX 21 IN INPUT O en 35 OEY 22 IN INPUT O en 39 BANK_3A 27 OUT OUTPUT3 O en 45 BANK_4A 31 OUT OUTPUT3 O en	19	BANK_1B		6	OUT	OUTPUT3	0	en		
27 BANK_2B 10 OUT OUTPUT3 0 en 28 FBKA_P 15 IN INPUT 0 en 31 REFA_P 18 IN INPUT 0 en 34 OEX 21 IN INPUT 0 en 35 OEY 22 IN INPUT 0 en 37 BANK_3A 27 OUT OUTPUT3 0 en 39 BANK_3B 26 OUT OUTPUT3 0 en 45 BANK_4A 31 OUT OUTPUT3 0 en +	25	BANK_2A		11	OUT	OUTPUT3	0	en		
28 FBKA_P 15 IN INPUT 0 en 31 REFA_P 18 IN INPUT 1 en 34 OEX 21 IN INPUT 0 en 35 OEY 22 IN INPUT 0 en 37 BANK_3A 27 OUT OUTPUT3 0 en 39 BANK_3B 26 OUT OUTPUT3 0 en 45 BANK_4A 31 OUT OUTPUT3 0 en	27	BANK_2B		10	OUT	OUTPUT3	0	en		
31 REFA_P 18 IN INPUT 1 en 34 OEX 21 IN INPUT 0 en 35 OEY 22 IN INPUT 0 en 37 BANK_3A 27 OUT OUTPUT3 0 en 39 BANK_3B 26 OUT OUTPUT3 0 en 45 BANK_4A 31 OUT OUTPUT3 0 en	28	FBKA_P		15	IN	INPUT	0	en		
34 OEX 21 IN INPUT 0 en 35 OEY 22 IN INPUT 0 en 37 BANK_3A 27 OUT OUTPUT3 0 en 39 BANK_3B 26 OUT OUTPUT3 0 en 45 BANK_4A 31 OUT OUTPUT3 0 en	31	REFA_P		18	IN	INPUT	1.000	en		
35 0EY 22 IN INPUT 0 en 37 BANK_3A 27 OUT OUTPUT3 0 en 39 BANK_3B 26 OUT OUTPUT3 0 en 45 BANK_4A 31 OUT OUTPUT3 0 en +	34	OEX		21	IN	INPUT	0	en		
37 BANK_3A 27 OUT OUTPUTS O en 39 BANK_3B 26 OUT OUTPUTS O en en 45 BANK_4A 31 OUT OUTPUTS O en en	35	UEY DANK 24		22	IN		0	en		l
45 BANK_4A 31 OUT OUTPUT3 0 en +	3/	BANK_3A		2/			0	en		1
	45	BANK_3B		20			0	en		
	4	DAME TA		1 11	001	SOIFOID	0	en		

Set Write View

In the **Set Write** view, the current data write register can be initialized and single data register bits can be set.

If the *Set and Run* mode is disabled (**BSDL.SetAndRun OFF**), the data write register is only prepared. To apply these settings to the boundary scan chain a **BSDL.RUN DR** command is required.

With *Set and Run* mode enabled, each modification of the data write register will be immediately transferred to the boundary scan chain.



The <i>Init DR / Init BSR</i> buttons ignores the filter settings, i.e. all data register bits are set to the specified value.

Set Read View

In the Set Read view, the expected values for the data read register can be verified and modified.

B::BSDL.	SET 3. iguration - ispPAC	C_CLK561	LOAV_XX	T48				-	E X
Instruction BYPASS SAMPLE PRELOAL EXTEST	ons	Data © bin • he:	format - 1 x	DR moo Samp Set V Set F	de Filte ple Vrite O Read B	er data uput 🔽 utput 🕅 : idi 🔽 :	Observe Intern Spotlight	Init BSR ExpectX ExpectH ExpectL	
Data re	gister: BOUNE	DARY (E	xpect	data, two	step DR m	ode, set	& run	mode enabled)	
1 2 3 4 5 6 7 9 11 17 19 25 27 28 31 34 35 37 39 39 45	DOTE LOCK SGATE RESET GOE PS1 PS0 PLL_BYPASS BANK_0A BANK_0A BANK_1A BANK_1A BANK_1A BANK_2A BANK_2A BANK_3A BANK_3A BANK_4A		pin 34 40 41 42 43 44 47 3 2 7 6 11 105 18 21 22 27 26 31	DITCYDE OUT IN IN IN IN UN OUT OUT OUT OUT OUT IN IN IN IN IN OUT OUT OUT OUT	UNPUT INPUT INPUT INPUT INPUT INPUT OUTPUT3 OUTPUT3 OUTPUT3 OUTPUT3 OUTPUT3 INPUT INPUT INPUT INPUT INPUT OUTPUT3 OUTPUT3 OUTPUT3 OUTPUT3 OUTPUT3	EXPECT H H H H H H H H H H H H H		H L (FAIL) H (PASS) L (PASS) H (PASS) L (PASS) L L L L L L L L L L L L L	E

In the *Last Result* column, the sampled results of the last read operation and the check results can be viewed.

The toggle button **File Info** / **Test Results** switches between the data register view and the file information view. When file information view is selected, the read data from the BSDL file of the IC can be viewed.

B::BSDL.SET 3.	
BSDL Configuration - ispPAC_CLK5610AV_XXT48	on
SAMPLE Show Statistic Finder Funder	
PRELOAD	
EXTEST V Set Read	
PSN Filoipfo	
num port pin pintype function cell safe ccell disval rslt	
General	
Filename : ispCLOCK5610Av_isc.bsm	
Entity : ISPPAC_CERSOIDAV_XI40	
Test Access Port Max. Tclk 25.0MHz, TCK Halt on LOW and HIGH	
TDI port : TDI	
TMS port : TMS	
TRST port :-	
Instruction Register (8 bits)	
BYPASS : supported	
SAMPLE : supported PRELOAD : supported	
EXTEST : supported	
CLAMP : supported	
IDCODE : supported	
INTEST : supported	
ISC_ADDRESS_INIT : not supported	
ISC_ADDRESS_SHIFT : not supported	
ISC_DIALA_SHIFF : not supported ISC_DISABLE : not supported	
ISC_DISCHARGE : not supported	
ISC_ENABLE : NOT SUPPORTED	
ISC_ERASE_DONE : not supported	
ISC_NOOP : not supported TSC_PRIGRAM : not supported	
ISC_PROGRAM_DONE : not supported	
ISC_PROGRAM_SECURITY: not supported TSC_READ	
LSC_PROGRAM_USERCODE: not supported	
LSC_USER_LOGIC_RESET: not supported	
Package : TQFP_48	
Boundary Scan Register (56 bits)	
1 LOCK 34 OUT OUTPUT3 BC_1 X 0 0 Z	
2 SGATE 40 IN INPUT BC_2 X	
$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	
5 PS1 43 IN INPUT BC_2 X	

For interactive tests of the PCB's interconnects, the following steps should be executed:

- 1. System setup
- 2. Configure boundary scan chain
- 3. Check boundary scan chain
- 4. Configure run mode
- 5. Execute level and connection tests

It is recommended to bring the system in down state. This avoids interferences from the debugger to the boundary scan tests. The following example will do the first 4 steps:

```
;System setup
SYStem.Down
SYStem.JtagClock 20MHz
BSDL.RESet
;Configure boundary scan chain
BSDL.FILE ispCLOCK5610Av_isc.bsm
BSDL.FILE ispCLOCK5610Av_isc.bsm
BSDL.FILE ispCLOCK5610Av_isc.bsm
BSDL.FILE LCMXCO1200C_ftBGA256.bsdl
;Check boundary scan chain
BSDL.HARDRESET
BSDL.SOFTRESET
IF !BSDL.CHECK.BYPASS()
(
    BSDL.BYPASSall
    PRINT %ERROR "Bypass test failed"
    ENDDO
)
IF !BSDL.CHECK.IDCODE()
(
    BSDL.IDCODEall
    PRINT %ERROR "ID code test failed"
    ENDDO
)
;Configure run mode
BSDL.SetAndRun ON
BSDL.TwoStepDR ON
```

;Execute level and connection tests....

For interactive tests, the two commands **BSDL.SetAndRun** and **BSDL.TwoStepDR** are useful. When enabled:

- A BSDL.RUN DR command will be automatically executed when a data register bit is modified (BSDL.SetAndRun ON)
- Each BSDL.RUN DR command will be execute two DR-SCAN operations (BSDL.TwoStepDR ON)

Set and Run Mode

With the *Set and Run* mode enabled (**BSDL.SetAndRun ON**), any change of a data register bit will be immediately executed, no matter if it was modified on the command line or interactively in a **BSDL.SET** window.

As a result, each modification of a data register bit or data register bit slice will immediately do the following:

- 1. Execute an *IR-SCAN* operation if any instruction register in the boundary scan chain was modified.
- 2. Execute a *DR-SCAN* operation.

A single *DR-SCAN* operation will first capture the data read register and then apply the new data write register:



To see the effect of the new write data on the sampled port state a second *DR-SCAN* operation is required. With the "*Two Step DR* mode" enabled (**BSDL.TwoStepDR ON**), this second *DR-SCAN* operation is executed automatically.

BSDL.TwoStepDR OFF	BSDL.TwoStepDR ON
Capture n-1	Capture n-1
Shift in n Shift out n-1	Shift in n Shift out n-1 (will be discarded)
Update to n	Update to n
	Capture n
	Shift in n Shift out n
	Update to n
Result: Boundary write registers in state n Result in read register window: n-1	Result: Boundary write registers in state n Result in read register window: n

Level Tests

The current signal levels can be checked by executing SAMPLE mode. There are several ways to do this:

- Set all required ICs in SAMPLE mode:
 - Select SAMPLE from the context menu for each IC in the BSDL.state window or
 - Select **SAMPLE** in the instruction list box in the **BSDL.SET** window for each required IC and click the **RUN** button on tab **Run** of the **BSDL.state** window.
- Click the SAMPLEall button on the Check tab of the BSDL.state window.

The results can be observed in the **BSDL.SET** windows of the respective ICs (*DR mode: Sample*).

—										
	B::BSDL.S	SET 3.								23
B::BSDL.state				• ×	В				📃 🫓 File In	ıfo
Configure Check	Run]			- DR m	ode — Filt	ter data —	Observe		
BYPASSall Test PASS SAM IDCODEall Test PASS	IPLEall T	'est done M ⊤	lax.TCK: 25 CK ok.	.0MHz	© Set	Write 🗹 (Read 📝 6	Dutput 🔲 Bidi 🔽	Intern Spotlight		
No. Entity Ins	truction	DR Name	DR	Size						
2 ispPAC_CLK5610AV_X+ SAM	PLE	BOUNDAR	Y I	56	intype	function	Reg.	Enable		
3 ispPAC_CLK5610AV_X≯ SAM	PLE	BOUNDAR	Y	56	N	INPUT	ŏ	en		^
	FLL	BOONDAN		727	UT	OUTPUT3	1	en		
at				-	UT	OUTPUT3	1	en en		_
				► a	ŬŤ	OUTPUT3	ī	en		
	25 B	BANK_ZA		10	OUT	OUTPUT3	1	en		
B::BSDL.SET 4.	27 B	FBKA P		15	IN	INPUT	0	en		
	31 R	REFA_P		18	IN	INPUT	1	en		≡
BSDL Configuration - LCMX012000	34 0	DEX		21	IN	INPUT	0	en		
	37 B	BANK 3A		27	OUT	OUTPUT3	1	en		
Instructions Dat	39 B	BANK_3B		26	OUT	OUTPUT3	1	en		
BYPASS A Ob	45 B	BANK_4A		31	OUT	OUTPUT3	1	en		
SAMPLE	47 B	SANK_4B		50	001	UUIPUIS	1	en		b d
PRELOAD										·
EXTEST		Set Red		JI 💌 ;	Spouignu					
Data register: BOUNDARY (s	ample)	intuno f	unction	Peg	Enable			-		
147 PT5F	B7 IN	NOUT B	IDIR	0	dis					
149 PT6A	A6 II	NOUT B	IDIR	1	dis					
151 PT6B	A7 I	NOUT B	IDIR	1	dis					
155 PT6D	C8 T	NOUT B		0	dis					
157 PT6E	D8 I	NOUT B	IDIR	ŏ	dis					
159 PT6F	D7 I	NOUT B	IDIR	0	dis					
161 PI/A 163 PT78				1	dis					
165 PT7C	AIO I	NOUT B	IDIR	ĭ	dis					
167 PT7D	A9 I	NOUT B	IDIR	0	dis			-		
] <							•			

The *Reg.* column in the data area of the **BSDL.SET** window shows the sampled register value, the *Enable* column shows the enable state of the register bit:

- Register function *OUTPUT2* or *OUTPUT3*:
 en: Output driver is enabled
 dis: output driver is disabled
- Register function *BIDIR*:

en: Output driver is enabled, cell operates as a driver (output)

dis: Output driver is disabled, cell operates as a receiver (input)

• Other register functions are always en

Connection Tests

Connection between ICs with boundary scan registers can be tested, by setting the signal driving IC into *EXTEST* mode and the signal receiving IC into *SAMPLE* mode.



Caution:

To avoid the risk of damaging the board, the boundary scan register of any signal driving IC (*EXTEST* mode) must be set to a safe state before applying any register settings, e.g. with **BSDL.SET** *<chip_number>* **BSR** * **SAFE**. Be sure to initialize all driving ICs <u>before</u> enabling the *Set and Run* mode.

The recommended preparation procedure for interactive connection tests is:

- 1. Disable the Set and Run mode for the first initialization of EXTEST mode.
- 2. Set the required ICs into SAMPLE or EXTEST mode.
- 3. Initialize all ICs in *EXTEST* mode to a safe state.
- 4. Enable the *Two step DR* mode.
- 5. Enable the *Set and Run* mode.
- 6. Opens the **BSDL.SET** windows of the required ICs.
- 7. Set DR mode of the receiving ICs (SAMPLE mode) to Sample.
- 8. Set DR mode of the driving ICs (EXTEST mode) to Set Write.
- 9. Set *Filter data* as required.

After this preparation procedure, the connections can be tested. In the following example, there exists a direct connection from port PL7A of IC4 to port PS1 of IC3:



For automated tests of the PCB's interconnects, a script with the following steps is required:

- 1. System setup
- 2. Configure boundary scan chain
- 3. Check boundary scan chain
- 4. Prepare boundary scan chain
- 5. Run tests

It is recommended to bring the system in down state, to avoid interferences from the debugger to the boundary scan tests. The following example will do the first 3 steps:

;System setup

SYStem.Down SYStem.JtagClock 20MHz BSDL.RESet BSDL.TwoStepDR OFF BSDL.SetAndRun OFF

;Configure boundary scan chain

BSDL.FILE ispCLOCK5610Av_isc.bsm BSDL.FILE ispCLOCK5610Av_isc.bsm BSDL.FILE ispCLOCK5610Av_isc.bsm BSDL.FILE LCMXC01200C_ftBGA256.bsdl

;Check boundary scan chain

```
BSDL.HARDRESET
BSDL.SOFTRESET
IF !BSDL.CHECK.BYPASS()
(
    BSDL.BYPASSall
    PRINT %ERROR "Bypass test failed"
    ENDDO
)
IF !BSDL.CHECK.IDCODE()
(
    BSDL.IDCODEall
    PRINT %ERROR "ID code test failed"
    ENDDO
)
;prepare boundary scan tests...
;run tests...
```

NOTE: The *Two Step DR* mode and the *Set and Run* mode must be turned off for automated tests (**BSDL.TwoStepDR OFF, BSDL.SetAndRun OFF**).

The preparation of the boundary scan chain includes the following steps:

- Set all ICs, which are not required for the board test, to BYPASS mode
- Set receiving ICs to SAMPLE mode and initialize to expected value X
- Set driving ICs in PRELOAD mode
- Initialize driving ICs to a safe state
- Enable result checking

These steps can be programmed as follows:

```
;Prepare boundary scan chain:
;Set unused ICs in BYPASS mode, set receiving IC in SAMPLE mode
BSDL.SET 1. IR BYPASS
BSDL.SET 2. IR BYPASS
BSDL.SET 3. IR SAMPLE
BSDL.SET 3. BSR * ExpectX
;Initialize signal driving IC
BSDL.SET 4. IR PRELOAD
BSDL.SET 4. BSR * SAFE
;Enable result checking
BSDL.CHECK ON
```

A test consists of individual test cycles. A test cycle will sample the results from the previous cycle and apply the driving signals for the next cycle:

All **BSDL.SET** *<chip_number>* **PORT** *<port_name>* commands are incremental, i.e. if a port is set to e.g. *H* in one cycle and is not modified in the next cycles, this port will always be checked for high signal level. To turn off the signal level check for a certain port, use this command: **BSDL.SET** *<chip_number>* **PORT** *<port_name>* **X**

In automated board test scripts it is recommended to use the command **BSDL.SET** <*chip_number>* **PORT** <*port_name>* <*value>* to modify the boundary scan registers. But the command **BSDL.SET** <*chip_number>* **BSR** <*bitslice>* <*value>* can also be used.

If a cycle runs without errors, this message will be printed to the **AREA** window for each IC which has set expected values:

Check IC <n> BOUNDARY: PASS

If an error was found, an error message is printed:

Check IC <n> BOUNDARY: FAIL

The execution of the automated board test script stops after the first error.

Full Example

In the following example two direct unidirectional connections from port PL7A of IC4 to port PS1 of IC3 and from port PL6D of IC4 to port PLL_BYPASS of IC3 will be tested.

```
;Example for PCB Test
;Check unidirectional connection IC4.PL7A -> IC3.PS1
;Check unidirectional connection IC4.PL6D -> IC3.PLL BYPASS
;
;System setup
SYStem.Down
SYStem.JtaqClock 20MHz
BSDL.RESet
BSDL.TwoStepDR OFF
BSDL.SetAndRun OFF
;Configure boundary scan chain
BSDL.FILE ispCLOCK5610Av isc.bsm
BSDL.FILE ispCLOCK5610Av isc.bsm
BSDL.FILE ispCLOCK5610Av isc.bsm
BSDL.FILE LCMXCO1200C ftBGA256.bsdl
;Check boundary scan chain
BSDL.HARDRESET
BSDL.SOFTRESET
IF !BSDL.CHECK.BYPASS()
(
    BSDL.BYPASSall
    PRINT %ERROR "Bypass test failed"
    ENDDO
)
IF !BSDL.CHECK.IDCODE()
(
    BSDL.IDCODEall
    PRINT %ERROR "ID code test failed"
    ENDDO
)
; Prepare boundary scan chain:
;Set unused ICs in BYPASS mode
BSDL.SET 1. IR BYPASS
BSDL.SET 2. IR BYPASS
;Initialize signal receiving IC
BSDL.SET 3. IR SAMPLE
BSDL.SET 3. BSR * ExpectX
;Initialize signal driving IC
BSDL.SET 4. IR PRELOAD
BSDL.SET 4. BSR * SAFE
;Enable result checking
BSDL.CHECK ON
```

;Cycle 0: First cycle -> no expected values, switch drivers to EXTEST ;Prepare driver: drive 00 BSDL.SET 4. PORT PL7A 0 BSDL.SET 4. PORT PL6D 0 BSDL.RUN ;Execute test: Apply initial driver settings BSDL.SET 4. IR EXTEST BSDL, RUN TR ;Cycle 1 PRINT "Check connections @Low" ; Prepare receiver: expect Low Low BSDL.SET 3. PORT PS1 T. BSDL.SET 3. PORT PLL BYPASS L ;Prepare driver: drive 11 BSDL.SET 4. PORT PL7A 1 BSDL.SET 4. PORT PL6D 1 ;Execute test: Sample cycle 0 results, apply cycle 1 settings BSDL.RUN DR ;Cvcle 2 PRINT "Check connections @High" ; Prepare receiver: expect High High BSDL.SET 3. PORT PS1 Η BSDL.SET 3. PORT PLL BYPASS H ;Prepare driver: drive 01 BSDL.SET 4. PORT PL7A 0 BSDL.SET 4. PORT PL6D 1 ;Execute test: Sample cycle 1 results, apply cycle 2 settings BSDL.RUN DR ;Cycle 3 PRINT "Check connections alternating @Low@High" ; Prepare receiver: expect Low High BSDL.SET 3. PORT PS1 T. BSDL.SET 3. PORT PLL BYPASS H ;Prepare driver: drive 10 BSDL.SET 4. PORT PL7A 1 BSDL.SET 4. PORT PL6D 0 ;Execute test: Sample cycle 2 results, apply cycle 3 settings BSDL.RUN DR ;Cycle 4: Last cycle -> no new drive signals PRINT "Check connections alternating @High@Low" ; Prepare receiver: expect High Low BSDL.SET 3. PORT PS1 Η BSDL.SET 3. PORT PLL BYPASS L ;Execute test: Sample cycle 3 results BSDL.RUN DR ;Test finished, print PASS result PRINT "" PRINT "Test Done (PASS)." ENDDO

Non-boundary scan devices can be tested, when all inputs can be controlled by boundary scan enabled devices and all outputs are connected to boundary scan enabled devices.



The following example will test the NAND gate and the D-flip-flop:

```
; System setup
SYStem.Down
BSDL.RESet
; Configuration of the boundary scan chain
BSDL.FILE IC1.bsdl ; BSDL file fo IC1
BSDL.FILE IC2.bsdl ; BSDL file fo IC2
; Check boundary scan chain
BSDL, SOFTRESET
IF !BSDL.CHECK.BYPASS()
(
   BSDL.BYPASSall
   PRINT %ERROR "Bypass test failed"
   ENDDO
)
IF !BSDL.CHECK.IDCODE()
(
   BSDL.IDCODEall
   PRINT %ERROR "ID code test failed"
    ENDDO
)
;Prepare boundary scan chain:
;set driving IC in EXTEST, receiving IC in SAMPLE mode
BSDL.SET 1. IR EXTEST
BSDL.SET 2. IR SAMPLE
;Initialize signal driver and receiver
BSDL.SET 1. BSR * SAFE
BSDL.SET 2. BSR * ExpectX
;Enable result checking
BSDL.CHECK ON
;Test the NAND gate
PRINT "Test NAND gate"
;Prepare driver: drive A:0 B:0
BSDL.SET 1. PORT Port1_A 0
                                 ;Input A
BSDL.SET 1. PORT Port1_B 0
                                 ;Input B
BSDL.RUN
;Sample previous result (A:0 & B:0), drive A:0 B:1
PRINT " check 0 & 0"
BSDL.SET 2. PORT Port2 B ExpectH ;Output Q
BSDL.SET 1. PORT Port1_A 0
BSDL.SET 1. PORT Port1_B 1
BSDL.RUN
```

;Sample previous result (A:0 & B:1), drive A:1 B:0 PRINT " check 0 & 1" BSDL.SET 2. PORT Port2 B ExpectH BSDL.SET 1. PORT Port1 A 1 BSDL.SET 1. PORT Port1 B 0 BSDL.RUN ;Sample previous result (A:1 & B:0), drive A:1 B:1 PRINT " check 1 & 0" BSDL.SET 2. PORT Port2 B ExpectH BSDL.SET 1. PORT Port1_A 1 BSDL.SET 1. PORT Port1 B 1 BSDL.RUN ;Sample previous result (A:1 & B:1) PRINT " check 1 & 1" BSDL.SET 2. PORT Port2 B ExpectL BSDL.RUN ;Test NAND gate finished BSDL.SET 2. PORT Port2 B ExpectX ;Test the D-FlipFlop PRINT "Test D-FlipFlop" Prepare driver: D:1 Clock:0 R:1 ; BSDL.SET 1. PORT Port1 D 1 ;Input D BSDL.SET 1. PORT Port1_F 0 ;Clock BSDL.SET 1. PORT Port1_G 1 ;Input R BSDL.RUN ;Sample reset PRINT " check reset" BSDL.SET 2. PORT Port2_E ExpectH ;Output Q BSDL.SET 2. PORT Port2_F ExpectL ;Output ^Q BSDL.RUN ;do a clock cycle, output should not change BSDL.SET 1. PORT Port1 F 1 BSDL.RUN BSDL.SET 1. PORT Port1_F 0 BSDL.RUN ;set D to 0 and do a clock cycle, output should not change BSDL.SET 1. PORT Port1 D 0 BSDL.RUN BSDL.SET 1. PORT Port1_F 1 BSDL.RUN BSDL.SET 1. PORT Port1_F 0 BSDL.RUN ;release reset and test data PRINT " check D:0" BSDL.SET 1. PORT Port1 G 0 BSDL.RUN

;Input D already 0 from previous test ;do a clock cycle, output should change after rising clock BSDL.SET 1. PORT Port1 F 1 BSDL.RUN BSDL.SET 2. PORT Port2_E ExpectL BSDL.SET 2. PORT Port2_F ExpectH BSDL.SET 1. PORT Port1_F 0 BSDL.RUN PRINT " check D:1" ;set D:1 BSDL.SET 1. PORT Port1_D 1 BSDL.RUN ;do a clock cycle, output should change after rising clock BSDL.SET 1. PORT Port1_F 1 BSDL.RUN BSDL.SET 2. PORT Port2_E ExpectH BSDL.SET 2. PORT Port2_F ExpectL BSDL.SET 1. PORT Port1 F 0 BSDL.RUN ;Test D-FlipFlop finished BSDL.SET 2. PORT Port2 E ExpectX BSDL.SET 2. PORT Port2_F ExpectX BSDL.CHECK OFF

;Test finished, print PASS result PRINT "" PRINT "Test Done (PASS)." ENDDO This chapter describes how to:

- Use the boundary scan oscilloscope
- Access IC specific data registers.

Boundary Scan Oscilloscope

The following command can be used to enable a simple boundary scan oscilloscope:

```
BSDL.SET <chip_number> OPTION BSRHISTORY ON
for IC <chip_number>
```

The command stores up to 32 samples for each bit of the boundary scan register. To view the waveforms in the **BSDL.SET** window, select **Sample** in the DR mode box.

B::BSDL.SET 4.								
BSDL Configuration - LCMX01200C_XXF256								
Instructions Data format DR mode Filter data SAMPLE Dbin Sample Set Write Set Read Vertication Vertic								
Data regis	ter: BOUNDARY ((sample)						
num por	t	pin	pintype	function	Reg.	Enable	History	
68 -	_	-	-	CONTROL	1	en		•
69 PL/	В	G5	INOUT	BIDIR	1	dis		
/0 -			-	CONTROL	1	en		
/1 PL/	A	G4	INOUT	BIDIR	1	dis		
/2 -	_		-	CONTROL	1	en		
73 PL6	D	F1	TNOUL	BIDIR	1	dis		
74 -	~	-	THOUT	CONTROL	1	en		
75 PL	C	EL	TNOOL	BIDIK	1	dis		
77 016	D	0	TNOUT		1	en		-

Each time a **BSDL.RUN** or **BSDL.RUN DR** is command is executed, a sample is stored. The background color of the history area changes its color every 5 samples. The signal colors provide information about the IO direction of the port:

- Blue: port is in input mode.
- Red: port is in output mode.
- Gray: port is in Z state or register bit is an internal cell.

To clear the history area, disable the **BSRHISTORY** option:

BSDL.SET 3. OPTION BSRHISTORY OFF ;Disables the boundary scan scope

Each instruction defined in the BSDL file can be set and the associated data register can be read and written to. This is useful, for example, if the IC provides additional functionality or test functions via the JTAG interface. The BSDL file can only provide information about the instruction names, instruction codes and the name and size of associated data registers. The meaning of the data register bits must be obtained from other sources such as the chip manufacturer's documentation.



In the **BSDL.SET** window, general data register masks are provided:

B::BSDL.SET 3.	
BSDL Configuration - ispPAC_CLK5610AV_XXT48	📝 File Info
Instructions Data format DR mode USERCODE ISC_ADDRESS_INIT ISC_ADDRESS_SHIFT ISC_DATA_SHIFT T	
Data register: ISC_ADDRESS (read)	
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0	*
	the state

B::BSDL.SET 3.	23
BSDL Configuration - ispPAC_CLK5610AV_XXT48	0
Instructions Data format DR mode Init DR USERCODE ISC_ADDRESS_INIT ISC_ADDRESS_SHIFT ISC_DATA_SHIFT T	
Data register: ISC_ADDRESS (write, single step DR mode, set & run mode disabled)	Ξ.
0 0 1 0 2 1 3 1 4 0	*
5 0 6 0 7 0 8 0 9 0	
< >>	- *

On the command line the data register bits of a non-boundary scan register can be modified with the command:

BSDL.SET <chip_number> DR <bitslice> <setvalue>

Set

sitslice> of current data register to value <setvalue> for IC <chip_number>

When result checking is enabled (**BSDL.CHECK ON**) and the expected values are set for the data register, a result message is printed to the **AREA** window after each **BSDL.RUN** or **BSDL.RUN** DR operation:

Check IC<n> <data_registername>: PASS | FAIL

with

- <n>: Number of the IC
- <data_registername>: Name of the current data register

NOTE: If a script is used for result checking, script execution stops after the first fail.

Tips and Tricks

To improve the readability of the data area in the **BSDL.SET** window, you can use the command:

BSDL.SET <chip_number> OPTION MARKLINES ON E

Enables alternating line colors for IC <*chip_number>*

B::BSDL.SET 3.									
BSDL Cor	BSDL Configuration - ispPAC_CLK5610AV_XXT48								
- Instruc BYPAS SAMPL PRELO EXTES	tions 5 AD T	format - 1 X	DR moo Samp Set V Set F	de Filte ple I Ir Vrite I O Read I B	er data — Iput 🔽 utput 🕅 Idi 🔽	Observe Intern Spotlight			
Data r	egister: BOUND	DARY (s	ample)			Deer	E		
num	port	15/9/5/9/15/5	p1n	pintype	function	Reg.	Enable		
1	LUCK		24	TN	UUTPUTS	1	en	^	
2	SGATE		40	TN	INPUT	0	en		
3	COL		41		TNPUT	0	en		
	GUE DC1		42		TNDUT	1	en		
i i	PS1 DS0		11	TN	TNDUT	1	en		
7			47	TN	TNDUT	1	en		
á	PANK OA		7/		OUTDUT3	1	en		
11	PANK_OP		2	OUT	OUTPUT3	0	en		
17	BANK 1A		7	OUT	OUTPUT3	ŏ	en	-	
19	BANK 1B		6	OUT	OUTPUT3	ŏ	en	-	
25	BANK 2A		11	OUT	OUTPUT3	0	en		
27	BANK 2B		10	OUT	OUTPUT3	0	en		
28	FBKA P		15	IN	INPUT	0	en		
31	REFA P		18	IN	INPUT	1	en		
34	OEX		21	IN	INPUT	0	en		
35	OEY		22	IN	INPUT	Ő	en		
37	BANK_3A		27	OUT	OUTPUT3	Ö	en		
39	BANK_3B		26	OUT	OUTPUT3	0	en		
45	BANK_4A		31	OUT	OUTPUT3	0	en	-	