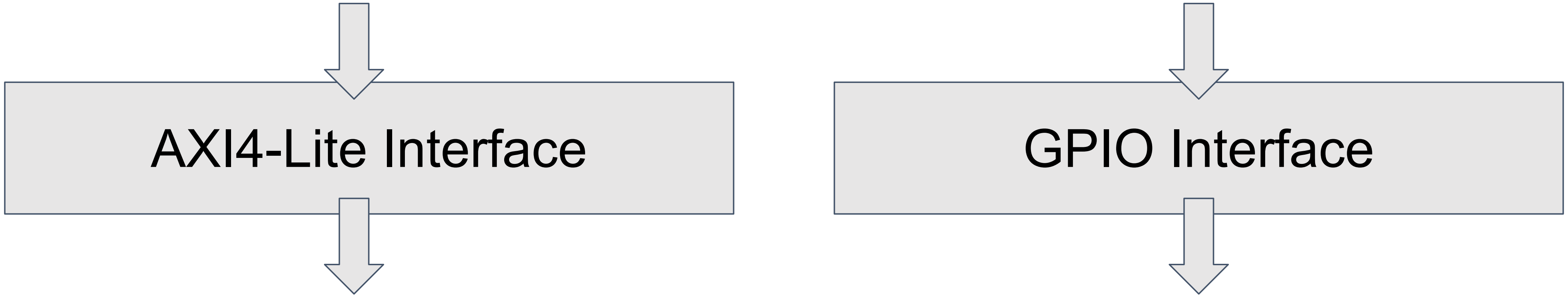
HLR: UVM based AXI-GPIO Verification Environment

Presenter:	Filza Shahid
Tech Lead:	Hassan Ashraf
Date:	14 Dec, 2023

TCP Overview

TCP Description

UVM based Verification Environment for Xilinx LogiCORE IP AXI-GPIO.



- Write Address
- Write Data
- Write Response
- Read Address
- Read Data & Response

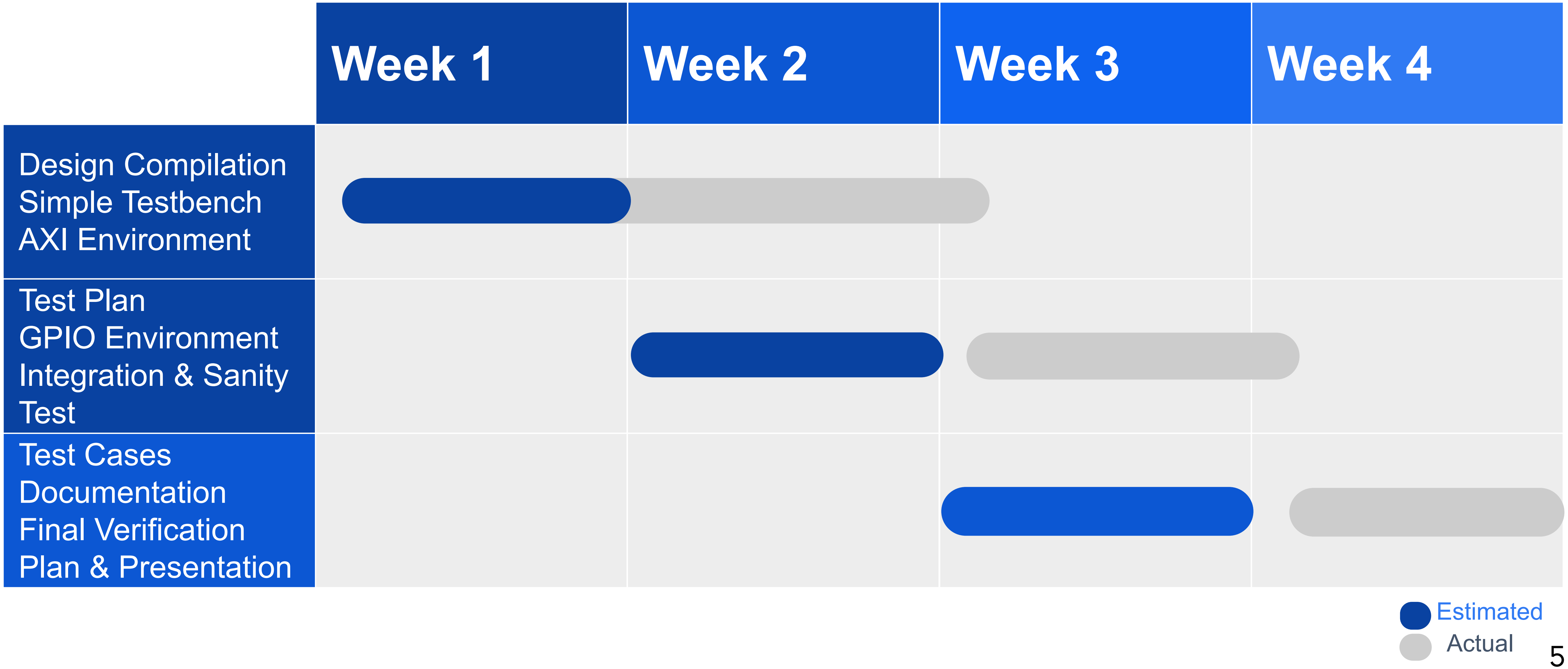
- Single/Dual Channel
- Independent Channel Width
- Input, Output & Direction ports
- Interrupt Generation

TCP Steps/Milestones

- Design Compilation
- Understanding AXI-IPIF and AXI-GPIO Specs
- Testing of Design with Simple Testbench
- Devising a Test Plan
- Building an AXI ENV and testing it
- Building GPIO ENV
- Integrating both ENVs
- Sanity Test
- Test Cases
- Documentation

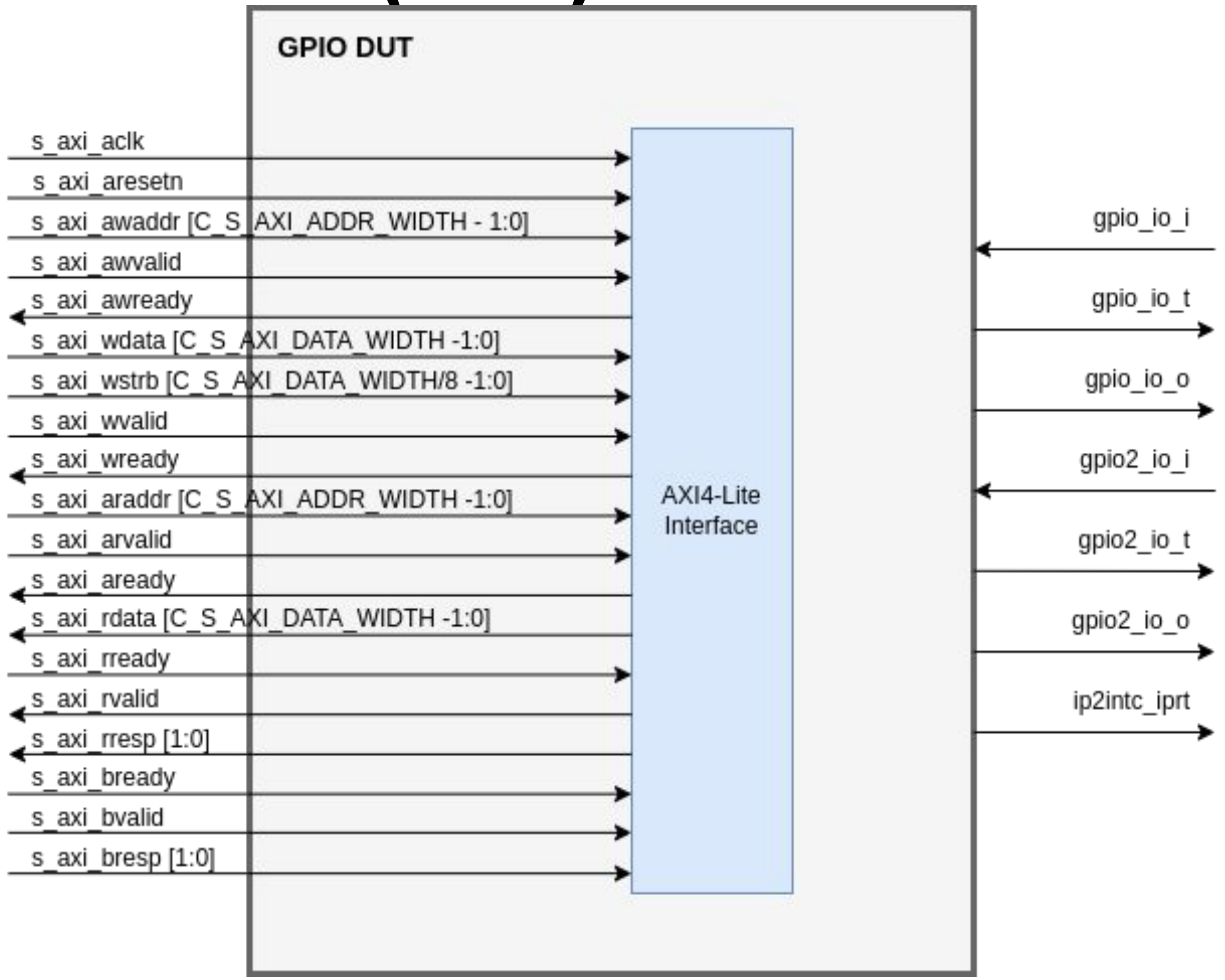
TCP Timeline

TCP Duration: November 10 2023 - December 7, 2023



Technical Overview

Design Under Test (DUT)



Design Under Test (DUT)

Signal Name	Interface	I/O	Description
s_axi_aclk	clk_rst_if	I	AXI Clock
s_axi_aresetn	clk_rst_if	I	AXI Reset, active-Low.
s_axi_awaddr	axi_intf	I	AXI Write address. The write address bus gives the address of the write transaction.
s_axi_awvalid	axi_intf	I	Write address valid. This signal indicates that valid write address and control information are available.
s_axi_awready	axi_intf	O	Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals.
s_axi_wdata	axi_intf	I	Write data.
s_axi_wstrb	axi_intf	I	Write strobes. This signal indicates which byte lanes to update in memory.

Design Under Test (DUT)

s_axi_wvalid	axi_intf	I	Write valid. This signal indicates that valid write data and strobes are available.
s_axi_wready	axi_intf	O	Write ready. This signal indicates that the slave can accept the write data.
s_axi_bresp	axi_intf	O	Write response. This signal indicates the status of the write transaction: “00” = OKAY “10” = SLVERR
s_axi_bvalid	axi_intf	O	Write response valid. This signal indicates that a valid write response is available.
s_axi_bready	axi_intf	I	Response ready. This signal indicates that the master can accept the response information.
s_axi_araddr	axi_intf	I	Read address. The read address bus gives the address of a read transaction.
s_axi_arvalid	axi_intf	I	Read address valid. When High, this signal indicates that the read address and control information is valid and remains stable until the address acknowledgement signal, S_AXI_ARREADY, is High.

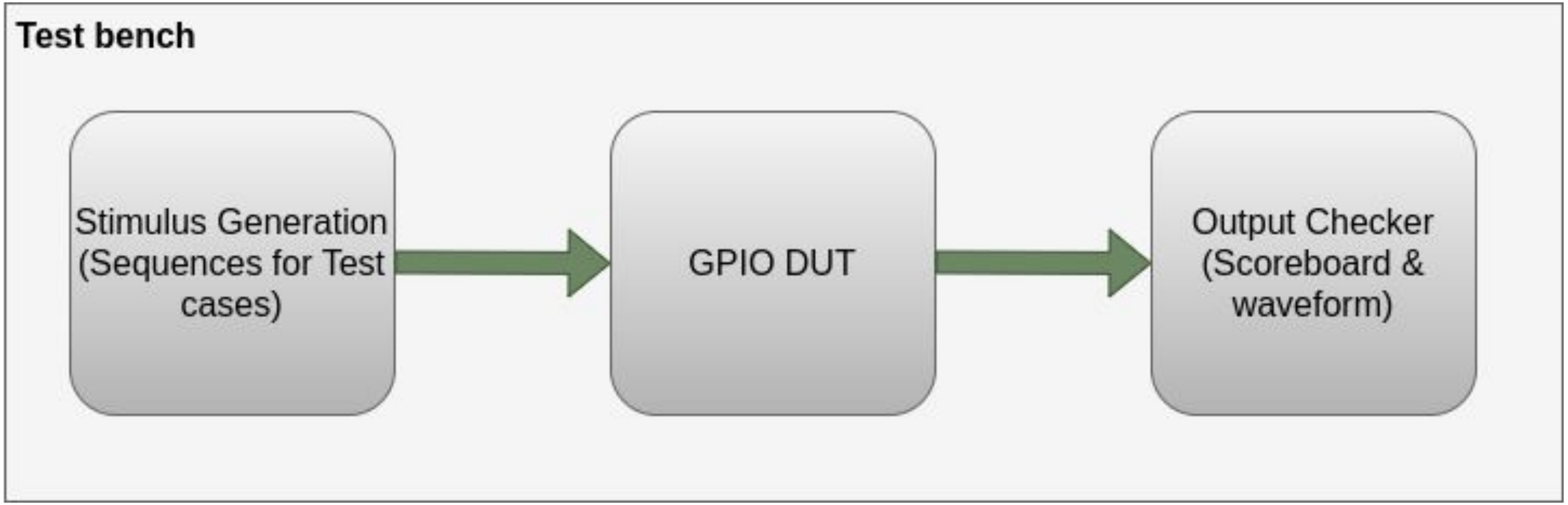
Design Under Test (DUT)

s_axi_arready	axi_intf	O	Read address ready. This signal indicates that the slave is ready to accept an address and associated control signals.
s_axi_rdata	axi_intf	O	Read data.
s_axi_rresp	axi_intf	O	Read response. This signal indicates the status of the read transfer.
s_axi_rvalid	axi_intf	O	Read valid. This signal indicates that the required read data is available and the read transfer can complete
s_axi_rready	axi_intf	I	Read ready. This signal indicates that the master can accept the read data and response information.
ip2intc_irpt	gpio_intf	O	AXI GPIO Interrupt. active-High, level sensitive signal.
gpio_io_i	gpio_intf	I	Channel 1 general purpose input pins.
gpio_io_o	gpio_intf	O	Channel 1 general purpose output pins.

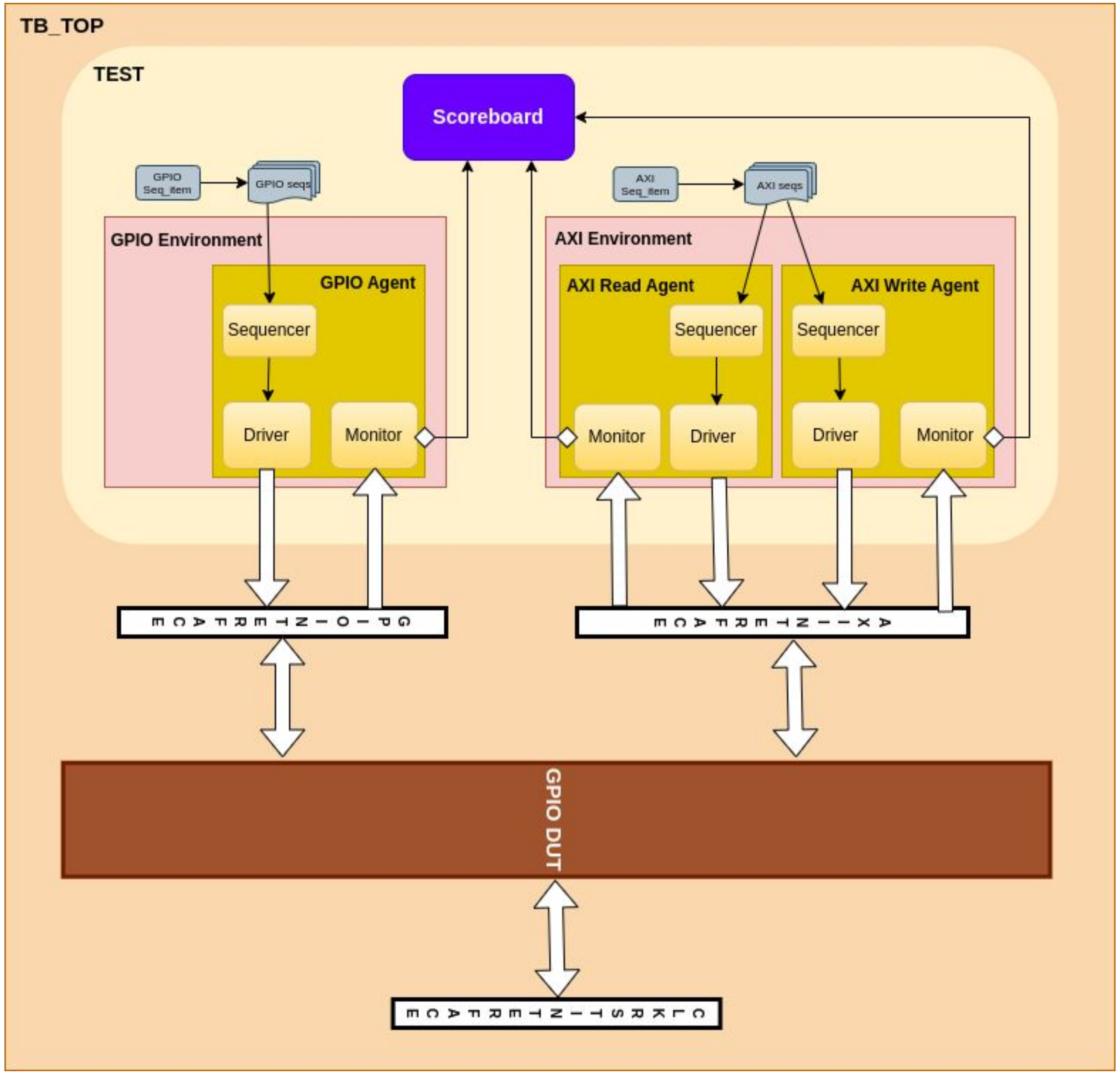
Design Under Test (DUT)

gpio_io_t	gpio_intf	O	Channel 1 general purpose 3-state pins.
gpio2_io_o	gpio_intf	O	Channel 2 general purpose output pins.
gpio2_io_i	gpio_intf	I	Channel 2 general purpose input pins.
gpio2_io_t	gpio_intf	O	Channel 2 general purpose 3-state pins

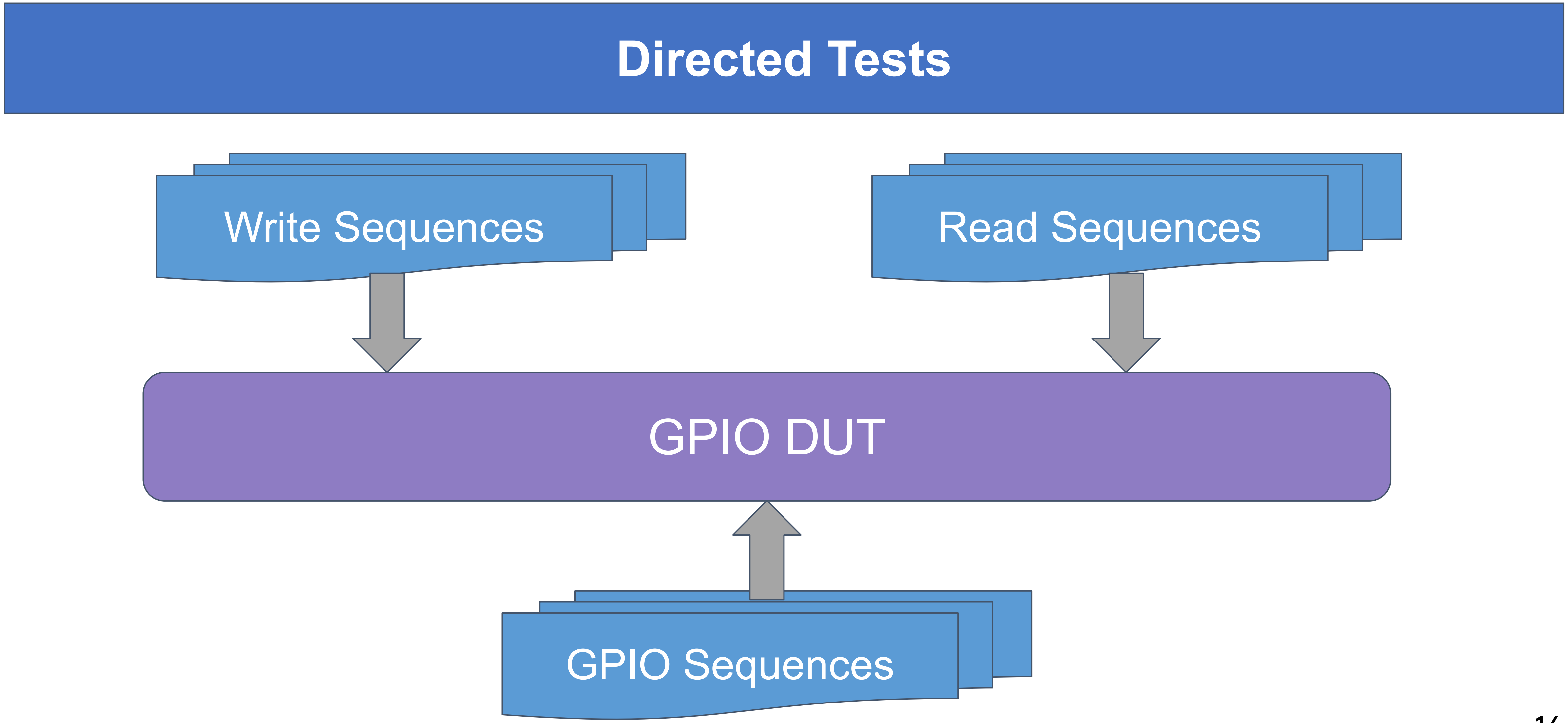
Block Diagram


Test bench
Stimulus Generation (Sequences for Test cases)
GPIO DUT
Output Checker (Scoreboard & waveform)

Testbench Structure



Verification approach



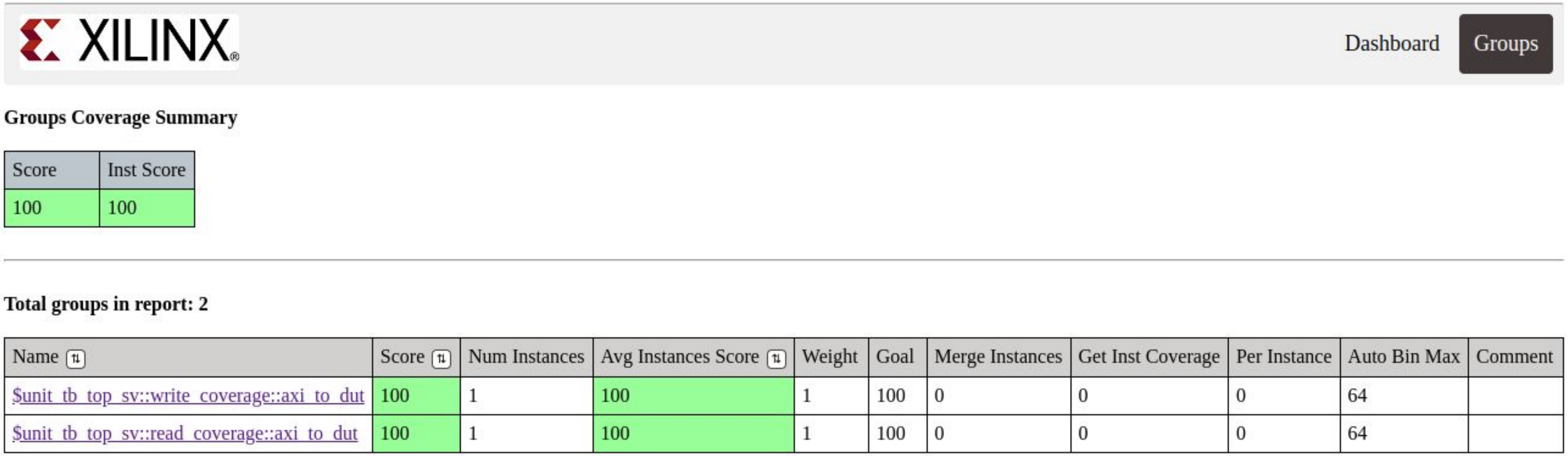
Statistics/Data points

Total Number of Test Cases = 11

Passed : 11

Failed : 0

Statistics/Data points



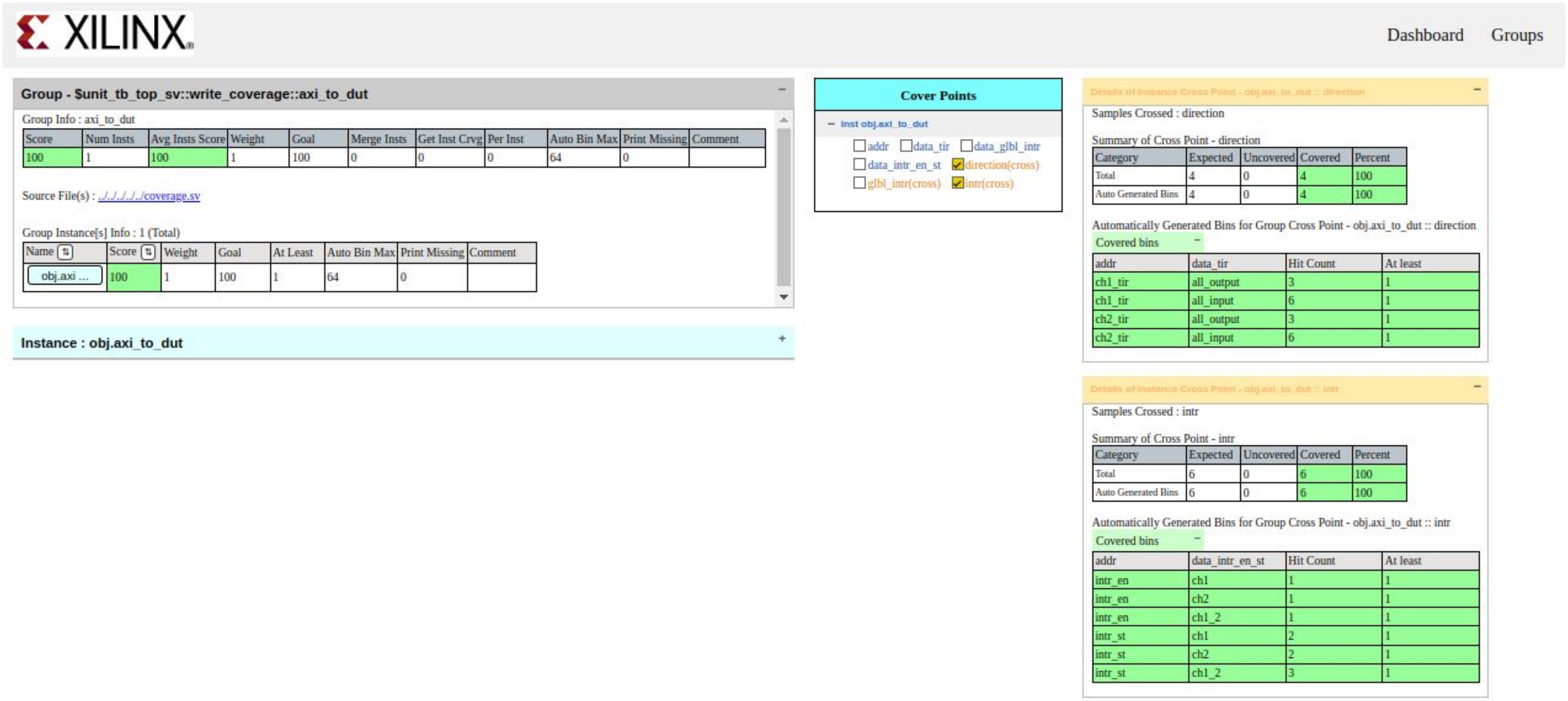
Groups Coverage Summary

Score	Inst Score
100	100

Total groups in report: 2

Name	Score	Num Instances	Avg Instances Score	Weight	Goal	Merge Instances	Get Inst Coverage	Per Instance	Auto Bin Max	Comment
$unit_tb_top_sv::write_coverage::axi_to_dut	100	1	100	1	100	0	0	0	64	
$unit_tb_top_sv::read_coverage::axi_to_dut	100	1	100	1	100	0	0	0	64	

Statistics/Data points



Group - $unit_tb_top_sv::write_coverage::axi_to_dut

Group Info : axi_to_dut

Score	Num Insts	Avg Insts Score	Weight	Goal	Merge Insts	Get Inst Crvg	Per Inst	Auto Bin Max	Print Missing	Comment
100	1	100	1	100	0	0	0	64	0	

Source File(s) : ../../../../../coverage.sv

Group Instance[s] Info : 1 (Total)

Name	Score	Weight	Goal	At Least	Auto Bin Max	Print Missing	Comment
obj.axi ...	100	1	100	1	64	0	

Instance : obj.axi_to_dut

Cover Points

− Inst obj.axi_to_dut

- ☐ addr
- ☐ data_tir
- ☐ data_glbl_intr
- ☐ data_intr_en_st
- ☑ direction(cross)
- ☐ glbl_intr(cross)
- ☑ intr(cross)

Details of Instance Cross Point - obj.axi_to_dut :: direction

Samples Crossed : direction

Summary of Cross Point - direction

Category	Expected	Uncovered	Covered	Percent
Total	4	0	4	100
Auto Generated Bins	4	0	4	100

Automatically Generated Bins for Group Cross Point - obj.axi_to_dut :: direction

Covered bins

addr	data_tir	Hit Count	At least
ch1_tir	all_output	3	1
ch1_tir	all_input	6	1
ch2_tir	all_output	3	1
ch2_tir	all_input	6	1

Details of Instance Cross Point - obj.axi_to_dut :: intr

Samples Crossed : intr

Summary of Cross Point - intr

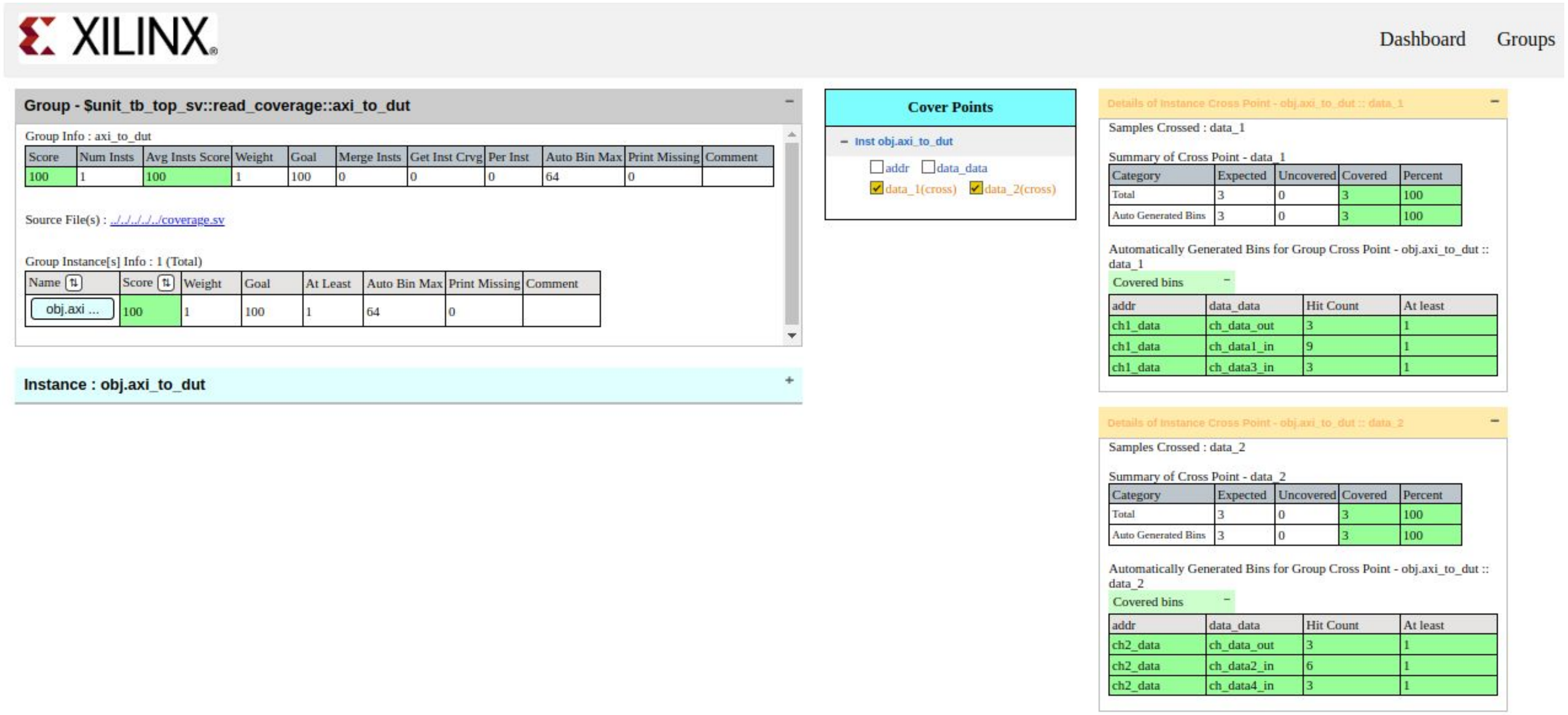
Category	Expected	Uncovered	Covered	Percent
Total	6	0	6	100
Auto Generated Bins	6	0	6	100

Automatically Generated Bins for Group Cross Point - obj.axi_to_dut :: intr

Covered bins

addr	data_intr_en_st	Hit Count	At least
intr_en	ch1	1	1
intr_en	ch2	1	1
intr_en	ch1_2	1	1
intr_st	ch1	2	1
intr_st	ch2	2	1
intr_st	ch1_2	3	1

Statistics/Data points



Dashboard Groups

Group - $unit_tb_top_sv::read_coverage::axi_to_dut

Group Info : axi_to_dut

Score	Num Insts	Avg Insts Score	Weight	Goal	Merge Insts	Get Inst Crvg	Per Inst	Auto Bin Max	Print Missing	Comment
100	1	100	1	100	0	0	0	64	0	

Source File(s) : ../../../../../coverage.sv

Group Instance[s] Info : 1 (Total)

Name	Score	Weight	Goal	At Least	Auto Bin Max	Print Missing	Comment
obj.axi ...	100	1	100	1	64	0	

Instance : obj.axi_to_dut

Cover Points

Inst obj.axi_to_dut

☐ addr ☐ data_data
☑ data_1(cross) ☑ data_2(cross)

Details of Instance Cross Point - obj.axi_to_dut :: data_1

Samples Crossed : data_1

Summary of Cross Point - data_1

Category	Expected	Uncovered	Covered	Percent
Total	3	0	3	100
Auto Generated Bins	3	0	3	100

Automatically Generated Bins for Group Cross Point - obj.axi_to_dut :: data_1

Covered bins

addr	data_data	Hit Count	At least
ch1_data	ch_data_out	3	1
ch1_data	ch_data1_in	9	1
ch1_data	ch_data3_in	3	1

Details of Instance Cross Point - obj.axi_to_dut :: data_2

Samples Crossed : data_2

Summary of Cross Point - data_2

Category	Expected	Uncovered	Covered	Percent
Total	3	0	3	100
Auto Generated Bins	3	0	3	100

Automatically Generated Bins for Group Cross Point - obj.axi_to_dut :: data_2

Covered bins

addr	data_data	Hit Count	At least
ch2_data	ch_data_out	3	1
ch2_data	ch_data2_in	6	1
ch2_data	ch_data4_in	3	1

Test Plan

Test Id	Test Name	Test Status	Test Description	Stimulus Generation Procedure	Checking Procedure	Comments
1	GPIO_ch_1_all_input	PASS	checks for GPIO channel 1 pins configured as input	Following sequences are executed in order: -> AXI write direction at channel 1 -> GPIO input sequence -> AXI read data at channel 1	Direction of channel 1 pins is set as input thorough AXI interface. Then a gpio sequence with input at channel 1 is executed through GPIO interface. Then the data at channel 1 is verified by reading through AXI interface.	It ensures that if the channel pins are configured as input, then the data obtained from reading these pins must be the input data passed to GPIO
2	GPIO_ch_1_all_output	PASS	checks for GPIO channel 1 pins configured as output	Following sequences are executed in order: -> AXI write direction at channel 1 -> AXI write data at channel 1 -> AXI read data at channel 1	Direction of channel 1 pins is set as output thorough AXI interface. Then a data is written to channel 1 through AXI interface that appears at GPIO channel 1 output. Then the data at channel 1 is verified by reading through AXI interface.	It ensures that if the channel pins are configured as output, then the data obtained from reading these pins must be the data that is written on to these pins thorugh AXI and is output of GPIO
3	GPIO_ch_2_all_input	PASS	checks for GPIO channel 2 pins configured as input	Following sequences are executed in order: -> AXI write direction at channel 2 -> GPIO input sequence -> AXI read data at channel 2	Direction of channel 2 pins is set as input thorough AXI interface. Then a gpio sequence with input at channel 2 is executed through GPIO interface. Then the data at channel 2 is verified by reading through AXI interface.	Same as 1
4	GPIO_ch_2_all_output	PASS	checks for GPIO channel 2 pins configured as output	Following sequences are executed in order: -> AXI write direction at channel 2 -> AXI write data at channel 2 -> AXI read data at channel 2	At first, direction of channel 2 pins is set as output thorough AXI interface. Then a data is written to channel 2 through AXI interface that appears at GPIO channel 2 output. Then the data at channel 2 is verified by reading through AXI interface.	Same as 2

Test Plan

5	GPIO_ch_1_2_input	PASS	checks for GPIO channel 1 & channel 2 pins both configured as input	Following sequences are executed in order: -> AXI write direction at channel 1 -> AXI write direction at channel 2 -> GPIO input sequence -> AXI read data at channel 1 -> AXI read data at channel 2	Direction of both channels pins is set as input thorough AXI interface. Then a gpio sequence with input at both channels is executed through GPIO interface. Then the data at both channels is verified by reading through AXI interface.	Same as 1
6	GPIO_ch_1_2_output	PASS	checks for GPIO channel 1 & channel 2 pins both configured as output	Following sequences are executed in order: -> AXI write direction at channel 1 -> AXI write direction at channel 2 -> AXI write data at channel 1 -> AXI write data at channel 2 -> AXI read data at channel 1 -> AXI read data at channel 2	Direction of both channels pins is set as output thorough AXI interface. Then a data is written to both channels through AXI interface that appears at GPIO output of both channels. Then the data at both channels is verified by reading through AXI interface.	Same as 2
7	GPIO_ch_1_input_2_output	PASS	checks for GPIO channel 1 pins configured as input & channel 2 pins configured as output	Following sequences are executed in order: -> AXI write direction at channel 1 -> AXI write direction at channel 2 -> AXI write data at channel 1 -> AXI write data at channel 2 -> GPIO input sequence -> AXI read data at channel 1 -> AXI read data at channel 2	Direction of both channels pins is set thorough AXI interface. Then a data is written to both channels through AXI interface. Also GPIO sequence is executed through GPIO interface. Then the data at both channels is verified by reading through AXI interface.	Although one channel is set as input and other is set as an output, data is is written to both channels and also input is driven to both channels too. At the end the test is verified by reading from both channels and getting the expected results.
8	GPIO_ch_1_output_2_input	PASS	checks for GPIO channel 1 pins configured as output & channel 2 pins configured as input	Following sequences are executed in order: -> AXI write direction at channel 1 -> AXI write direction at channel 2 -> AXI write data at channel 1 -> AXI write data at channel 2 -> GPIO input sequence -> AXI read data at channel 1 -> AXI read data at channel 2	Direction of both channels pins is set thorough AXI interface. Then a data is written to both channels through AXI interface. Also GPIO sequence is executed through GPIO interface. Then the data at both channels is verified by reading through AXI interface.	Same as 7

Test Plan

9	GPIO_ch_1_2_intr_en_with_input_at_ch_any	PASS	checks for GPIO channel 1 & channel 2 pins configured as input, with interrupt enabled for both channels	Following sequences are executed in order: -> AXI write Global interrupt enable -> AXI write Interrupt enable for both channels -> AXI write direction at channel 1 -> AXI write direction at channel 2 -> GPIO input sequence -> interrupt at channel 1 sequence -> interrupt at channel 2 sequence -> interrupt at both channels sequence	Both global and local interrupt registers are enabled for both channels. Then direction of both channels pins is set as input through AXI interface. GPIO sequence is executed to configure the pins to some initial value. Then interrupts sequences are executed and check whether interrupt appears at output or not. If appears, corresponding interrupt status register bit is set and data at both channels is verified by reading through AXI interface.	It ensures that the interrupt is detected only when the global interrupt register and the local interrupt register bit for corresponding channel is set. Once the interrupt is detected by change at the channel for which interrupt is enabled, it is necessary to set the corresponding channel bit in interrupt status register to get the next interrupt.
10	GPIO_ch_1_intr_en_with_input_at_ch_any	PASS	checks for GPIO channel 1 & channel 2 pins configured as input, with interrupt enabled for channel 1	Following sequences are executed in order: -> AXI write Global interrupt enable -> AXI write Interrupt enable for both channels -> AXI write direction at channel 1 -> AXI write direction at channel 2 -> GPIO input sequence -> interrupt at channel 1 sequence -> interrupt at channel 2 sequence -> interrupt at both channels sequence	Global and local interrupt register bit for channel 1 are enabled. Then direction of both channels pins is set as input through AXI interface. GPIO sequence is executed to configure the pins to some initial value. Then interrupts sequences are executed and check whether interrupt appears at output or not. If appears, corresponding interrupt status register bit is set and data at both channels is verified by reading through AXI interface.	Same as 9
11	GPIO_ch_2_intr_en_with_input_at_ch_any	PASS	checks for GPIO channel 1 & channel 2 pins configured as input, with interrupt enabled for channel 2	Following sequences are executed in order: -> AXI write Global interrupt enable -> AXI write Interrupt enable for both channels -> AXI write direction at channel 1 -> AXI write direction at channel 2 -> GPIO input sequence -> interrupt at channel 1 sequence -> interrupt at channel 2 sequence -> interrupt at both channels sequence	Global and local interrupt register bit for channel 2 are enabled. Then direction of both channels pins is set as input through AXI interface. GPIO sequence is executed to configure the pins to some initial value. Then interrupts sequences are executed and check whether interrupt appears at output or not. If appears, corresponding interrupt status register bit is set and data at both channels is verified by reading through AXI interface.	Same as 9

Demo

Demonstrating the Project

Challenges

- Design not being Compiled
- Dependency of IP and Simulator on precompiled libraries for Xilinx VHDL Design
- EDA tools
- Long simulation time and crash issues in Vivado

Thankyou.
Questions?