

User Manual for PCIe ISA Bus Controller



Table of Contents

1	INT	RODUCTION	4
	1.1	PURPOSE	4
	1.2	SCOPE	4
	1.5 1.4	EVALUATION BOARD AND CORE REQUIREMENTS	4 5
2	PCI	E TO ISA BUS CONTROLLER CORE	6
	2.1	BLOCK DIAGRAM	6
	2.2	DESCRIPTION	6
	2.3	PIN OUTS OF IW- PCIE ISA BRIDGE CORE	7
3	QUI	ICK START1	1
	3.1	CONNECTING TO A HOST COMPUTER	1
	3.1.	l Installation Requirements1	1
	3.1.2	2 Board Installation & Testing 1	1
	3.1.3	3 Procedure for Demo 1	2
A	PPENI	DIX A 2	6



List of Figures

Figure 1: Detailed view of iW-PCIe to ISA controller core	6
Figure 2: IO Write Cycle	
Figure 3: IO Read Cycle	20
Figure 4: Memory Write Cycle	
Figure 5: Memory Read Cycle	25

List of Tables

Table 1: Pin outs of iW-PCIe ISA Bridge

7



1 Introduction

1.1 Purpose

The purpose of this document is to explain the procedure to power-on and setting up working environment of the PCIe to ISA Bridge for demo purpose.

1.2 Scope

This document describes the Hardware connection procedure to power-on the board and establishes connection with the PC.

1.3 Features

PCIe Interface

- The Xilinx endpoint cores for PCIe follows PCI express base specification v1.1 layering model.
- o 32-bit internal data path
- The endpoint core implements the physical layer, datalink layer, transaction layer & configuration management layer.
- Six individually programmable BAR's & expansion ROM BAR.
- Supports MSI & INTX emulation.
- Supports removal of corrupt packets for error detection and recovery.
- Compatible with PCI/PCI Express power management functions.
- Used in conjunction with NXP PX1011A PCI Express standalone PHY to achieve high transceiver capability, 2.5 GBPS line speed, automatic clock and data recovery, 8b/10b encode and decode.
- Supports a maximum transaction payload of up to 512 bytes.

ISA Master Interface

- The ISA Bridge implements a 16-bit data interface.
- Supports Bus clock of 8 MHz for ISA interface.
- Supports a 20-bit system address lines tristate, which can be latched on to the falling edge of bus address latch enable signal.
- Supports latchable address lines, these unlatched address signals give the system up to 16 MB of address ability.



1.4 Evaluation Board and Core requirements

- o Spartan-3 PCI Express Kit
- Mother Board with PCIe slot with PCIe tree software installed
- PC/laptop with ChipScope software installed
- Endpoint core for PCI express, PIO Module from Xilinx
- PCIe to ISA bus controller core



2 PCIe to ISA Bus controller Core

2.1 Block Diagram



Figure 1: Detailed view of iW-PCIe to ISA controller core

2.2 Description

The PCIe Bridge has an endpoint PIPE v1.7 (PHY Interface) for PCIe 1 lane core from Xilinx, Programmed I/O module & ISA controller. The endpoint core from xilinx implements the physical layer (PHY interface), data link layer, transaction layer & configuration management layer of PCIe base specification v1.1 layering model. The PIO design interfaces with the endpoint for PCI Express core's transaction interface & responds with read/write transaction for memory or IO transaction from the endpoint core.

The ISA bus controller is implemented in user interface side of the PIO design. The host processor can access the unit through memory/IO read and write commands. The ISA bus is a 16bit interface, which can be used to connect peripheral components to the host CPU through ISA bus.



2.3 Pin outs of iW- PCIe ISA Bridge core

The pin outs of iW- PCIe ISA Bridge is as shown in the table below.

iW-PCIe ISA Bridge PINS	FPGA PINS
powerdown[0]	AF22
powerdown[1]	AD23
resetn	AF24
rxpolarity	AE24
txclk	AE21
txcompliance	AE23
txdata[0]	AD15
txdata[1]	AE15
txdata[2]	AF15
txdata[3]	AE19
txdata[4]	AF19
txdata[5]	AE20
txdata[6]	AF20
txdata[7]	AD21
txdatak[0]	AE22
txdetectrx_loopback	AF21
txelecidle	AF23
phystatus	AF12
rxdata[0]	AE8
rxdata[1]	AC7
rxdata[2]	AF6

Table 1: Pin outs of iW-PCIe ISA Bridge



iW-PCIe ISA Bridge PINS	FPGA PINS
rxdata[3]	AE6
rxdata[4]	AD6
rxdata[5]	AC6
rxdata[6]	AE5
rxdata[7]	AD5
rxdatak[0]	AF8
rxelecidle	AF4
rxstatus[1]	AD10
rxstatus[2]	AC11
rxvalid	AD12
rxclk	AE13
sys_reset_n	AE4
sa_o[0]	M3
sa_o[1]	J7
sa_o[2]	M7
sa_o[3]	J6
sa_o[4]	N7
sa_o[5]	H5
sa_0[6]	M8
sa_o[7]	H2
sa_0[8]	N8
sa_0[9]	J5
sa_o[10]	P8
sa_o[11]	J4



iW-PCIe ISA Bridge PINS	FPGA PINS
sa_o[12]	P2
sa_o[13]	K7
sa_o[14]	P7
sa_o[15]	K5
sa_o[16]	R1
sa_o[17]	L8
sa_o[18]	P1
sa_o[19]	L7
la_o[17]	R2
la_o[18]	H1
la_o[19]	R3
la_o[20]	L1
la_o[21]	T1
la_o[22]	L2
la_0[23]	T2
sbhe_n_o	L4
sd_io[0]	M19
sd_io[1]	P20
sd_io[2]	M20
sd_io[3]	T20
sd_io[4]	K20
sd_io[5]	P21
sd_io[6]	J20



iW-PCIe ISA Bridge PINS	FPGA PINS
sd_io[7]	R21
sd_io[8]	H20
sd_io[9]	P24
sd_io[10]	J21
sd_io[11]	P22
sd_io[12]	H21
sd_io[13]	R24
sd_io[14]	H22
sd_io[15]	R22
ior_n_o	V6
iow_n_o	U7
memr_n_o	W5
memw_n_o	V7
bclk_o	R8
bale_o	R7



3 Quick Start

3.1 Connecting to a Host computer

Follow the steps below to connect the Spartan-3 PCI Express board to the host computer through PCIe link to test the functionality of iW-PCIe ISA Bridge core.

3.1.1 Installation Requirements

The items listed below are necessary to install Spartan-3 PCI Express board to the host computer

• PC/laptop with Chipscope software installed.

• Host computer of windows NT/2000 or windows XP OS having an available PCIe slot, with installed PCIe Tree software.

3.1.2 Board Installation & Testing

- 1. Before connecting Spartan-3 PCI Express Kit in the PCIe slot check all these settings are properly done for starter kit
 - Select the master parallel mode for FPGA configuration by installing M2 in JP3 Header.
 - Other Jumpers position on Board, JP8 2-3, JP1 2-3, JP2 2-3, JP5 1-2, JP6 2-3, JP9, J4.
 - Select the power source from the PCIe edge connector for this install the fuse in socket F2 position (dont place separate fuse in F1 position).
- 2. Connect the Xilinx platform USB cable to the PC/laptop USB port from JTAG socket J2 of PCIe board for programming & to check the ISA waveforms on chipscope viewer, After this place the board in PCIe slot of a host computer.
- 3. Program the MCS file pcie_isa_bridge.mcs provided with user manual to the Spartan-3 PCI Express board, for this first program the on-board 8 Mb xilinx XCF08P parallel Platform Flash PROM then configure the FPGA from the image stored in the Platform flash PROM by power cycling (switch off & on the board).
- 4. Run the PCItree software on the host computer where the Spartan-3 PCI Express board is installed.
- 5. Check the software overview part to get more information regarding Pcitree software for read & write of memory & io space of host computer.



3.1.3 Procedure for Demo

- Connect Spartan-3 PCI Express board to the PCIe slot of host computer also connect the Xilinx platform USB cable to the PC/laptop in which chipscope software is installed.
- Start the PCItree software installed in the host computer to which Spartan-3 PCI Express board is connected, then Press OK



• The software will scan all the PCI bus attached to the host computer & displays all the PCI bus as the tree structure. Each PCI component has an integer number for bus, device and function (bdf).

hus dev	func:	ow INT routing highest EXIT
	÷lo P L	show Mem Map
- host CPU		
	Host/PCI; Bridge I	Host/PCI; Bridge Device
⊡.0.28.0	0->1 (2) PCI/PCI; Br:	DID: x2778 no device name found no
1.00.0	1->2 (2) PCI/PCI;	SubVID: x8086 Intel
0.28.4	0->3 (3) PCI/PCI; Br:	SubID: x348D no-name
3.00.0	PCI/ISA; Bridge	rev.: x00 no INT
0.28.5	0->4 (4) PCI/PCI; Br:	edit ConfReg: Nr of ConfRegs:
4.00.0	Ethernet; Netwo	□
4.00.3	16550 Compatibl	
4.00.4	o. serial bus D	use BIOS int
0.29.0	Universal Host Cor	Write ConfReg refresh
0.29.1	Universal Host Con	🗌 refr after wr. dump:
0.29.2	Universal Host Cor	
0.29.3	o seriel bus Dev	Config Space Dump: (type 1 xs)
E: 0.30.0	$0 \rightarrow 5$ (5) Subtractive	2778 8086 <00 : DID VID
5.04.0	VGA; PC Compati	2090 0106 <04 : Stat Cmd
5.05.0	Ethernet; Netwo	0600 0000 <08 : BaseClass SubClass F 0000 0000 <0C : BIST Header LatTimer
0.31.0	PCI/ISA; Bridge De	0000 0000 <10 : BAR 0
0.31.1	o. Mass Storage Co	0000 0000 <14 : BAR 1
0.31.3	SMBus; Serial Bus	0000 0000 <18 : BAR 2
		0000 0000 <1C : BAR 3
		0000 0000 <20 : BAR 4



• Locate the Spartan-3 PCI Express board in the PCI bus list, once you locate the device PCItree software will displays bus number, device number, function number, Vendor ID, device ID & configuration space contents in the right side of the pcitree window.

direct select: bus: dev: func: 3 3 0 0 0 bus: dev: func: 3 0 0 0 bus: dev: func: 3 0 0 0 bus: dev: func: 3 0 0 bus: dev: func: 3 0 0 bus: dev: func: 3 0 0 bus: dev: func: 3 0 0 CI/ISA; Bridge Device VID: x10EE Xilinx Corp 0 0.00 0 Host/PCI; Bridge I 0 0.28.0 0->1 (2) PCI/PCI; Bridge I 0 0.28.4 0->3 (3) PCI/PCI; Bridge 0 0.28.5 0->4 (4) PCI/PCI; Bridge 0 0.28.5 0->4 (4) PCI/PCI; Bridge 0 0.28.5 0->4 (4) PCI/PCI; Bridge 0 0.29.0 Universal Host Cor 0 0.29.1 Universal Host Cor 0 0.29.2 Universal Host Cor 0 0.29.3 Universal Host Cor 0 0.29.3 Universal Host Cor
□ 0.00.0 Host/PCI; Bridge I □ 0.00.0 Host/PCI; Bridge I □ 0.28.0 0->1 (2) PCI/PCI; Bridge I □ 1.00.0 1->2 (2) PCI/PCI; Bridge I □ 1.00.0 1->2 (2) PCI/PCI; Bridge I □ 0.28.4 0->3 (3) PCI/PCI; Bridge I □ 3.00.0 PCI/ISA; Bridge I SubVID: x10EE Xilinx SubID: x0007 no device name found no SubID: x0007 no-name □ 3.00.0 PCI/ISA; Bridge □ 0.28.5 0->4 (4) PCI/PCI; Bridee □ 4.00.0 Ethernet; Networe edit ConfReg: □ 0.29.0 Universal Host Core Nr of ConfRegs: □ 0.29.1 Universal Host Core Write ConfReg □ refr after wr. Config Space Dump: (type 1 xs)
-0.29.7 o. serial bus Dev: 0007 10KK <00 : DID VID -0.30.0 0->5 (5) Subtractive. 0007 10KK <00 : DID VID -5.04.0 VGA; PC Compati 0601 0000 <08 : BaseClass SubClass I -0.31.0 PCI/ISA; Bridge D: 0000 3001 <10 : BAR 0 io -0.31.1 o. Mass Storage C: 0000 0000 <14 : BAR 1 mem 32bit -0.31.3 SMBus; Serial Bus 0000 0000 <1C : BAR 3



• Select memory BAR register space to access host cpu memory space or select IO BAR register to access host cpu IO space. In this screenshot IO BAR (address 10h) is selected.

ň	PciTree		
	-direct select: hus: dev:	func: 😚 sh	ow INT routing highest E X I T
	· 북 3 · 북 0	국 원	show Mem Map
	E-bost CPII		
		Host/PCT: Bridge I	PCI/ISA; Bridge Device
	⊡ 0.28 0	0->1 (2) PCT/PCT: Br	VID: x10EE Xilinx Corp
	1.00.0	1->2 (2) PCI/PCI:	DID: X0007 no device name found no (
	⊡.0.28.4	0->3 (3) PCI/PCI; Br:	SubVID: XIOBE XIIINX SubID: X0007 no-neme
	3.00.0	PCI/ISA; Bridge	rev.: x00 xFF<-INTA#
	⊡ 0.28.5	0->4 (4) PCI/PCI; Br:	
	-4.00.0	Ethernet; Netwo	
	-4.00.3	16550 Compatibl	x00003001 hex 0 16 0 64
	4.00.4	o. serial bus D	use BIOS int
	0.29.0	Universal Host Cor	Write ConfReg vofwork
	0.29.1	Universal Host Cor	
	0.29.2	Universal Host Cor	refr after wr.
	0.29.3	Universal Host Cor	Config Space Dump: (type 1 xs)
	0.29.7	o. serial bus Dev:	0007 10WE <00 : DID MID
	Ė 0.30.0	$0 \rightarrow 5$ (5) Subtractive,	0010 0147 <04 : Stat Cmd
	- 5.04.0	VGA; PC Compati	0601 0000 <08 : BaseClass SubClass I
	5.05.0	Ethernet; Netwo	0000 0010 <oc :="" bist="" header="" lattimer<="" th=""></oc>
	0.31.0	PCI/ISA; Bridge De	0000 3001 <10 : BAR 0 io
	0.31.1	o. Mass Storage Co	4800 0000 <14 : BAR 1 mem 32bit
	····· 0.31.3	SMBus; Serial Bus	0000 0000 <18 : BAR 2
			0000 0000 <20 : BAR 3



• To write into the IO space of the host cpu, select IO BAR register & double click on that BAR regsiter & then press yes tab on the information window.



Tree PciTree		
direct select: bus: dev: 3 70	func:	ow INT routing highest E × I T busnr: 5 About
- host CPU - 0.00.0 - 0.28.0 - 1.00.0 - 0.28.4 - 3.00.0 - 0.28.5 - 4.00.0 - 4.00.3 - 0.29.0 - 0.29.1 - 0.29.2	Host/PCI; Bridge I 0->1 (2) PCI/PCI; Br: 1->2 (2) PCI/PCI; 0-> PCItree 0-> Are you ready I BAR read-write If not: a BARra Yes N	PCI/ISA; Bridge Device VID: x10EE Xilinx Corp DID: x0007 no device name found no · SubVID: x10EE Xilinx to perform e(0xfffffff)-read-restore ? nge of 1MByte is assumed lo Cancel
	o. serial bus Dev: 0->5 (5) Subtractive, VGA; PC Compati Ethernet; Netwo PCI/ISA; Bridge D¢ o. Mass Storage C¢ SMBus; Serial Bus	Conning Space Dump. (type 1 xs) 0007 10EE <00 : DID VID 0010 0147 <04 : Stat Cmd 0601 0000 <08 : BaseClass SubClass I 0000 0010 <0C : BIST Header LatTimer 0000 3001 <10 : BAR 0 io 4800 0000 <14 : BAR 1 mem 32bit 0000 0000 <18 : BAR 2 0000 0000 <1C : BAR 3 0000 0000 <20 : BAR 4



• The window containing all the registers under that BAR space are listed, we can access those register. To write into any register of IO space select some register, enter the data to be written in edit memory tab.

00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 000000000 <x00000004< td=""> 000000000<th></th><th>BAR space</th><th></th><th></th></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<>		BAR space		
00000000 -x00000004 00000000 -x00000004 00000000 -x00000010 000000000 -x00000000 000000000 -x00000000 000000000 -x000000000 000000000 -x000000000 000000000 -x000000000 000000000 -x000000000 000000000 -x000000000 00				
00000000 -x00000004 00000000 -x00000000 00000000 -x00000001 00000000 -x00000014 00000000 -x00000014 00000000 -x00000014 00000000 -x00000012 00000000 -x00000014 00000000 -x00000010 00000000 -x00000010 000000000 -x00000010 000000000 -x00000010 000000000 <th></th> <th>00000000</th> <th><x00000000< th=""><th>auto read memory OK</th></x00000000<></th>		00000000	<x00000000< th=""><th>auto read memory OK</th></x00000000<>	auto read memory OK
00000000 *200000000 00000000 *200000010 00000000 *200000010 00000000 *200000010 00000000 *200000010 00000000 *200000010 00000000 *200000010 00000000 *200000000 00000000 *200000000 00000000 *200000000 00000000 *200000000 00000000 *200000000 00000000 *200000000 00000000 *200000000 00000000 *200000000 00000000 *200000000 00000000 *200000000 00000000 *200000000 00000000 *200000000 00000000 *200000000 00000000 *200000000 00000000 *200000000 000000000 *200000000 000000000 *2000000000 000000000 *2000000000 000000000 *2000000000 000000000 *20000000000 000000000 *2000000000 000000000 *2000000000		00000000	≺x0000004	IO Space
00000000 -:x00000010 00000000 -:x00000110 00000000 -:x00000012 00000000 -:x00000014 000000000 -:x00000014		00000000	<x00000008< th=""><th>here : 00003000</th></x00000008<>	here : 00003000
00000000 <00000014 00000000 <00000012 00000000 <00000012 00000000 <00000012 00000000 <00000012 00000000 <00000012 00000000 <00000012 00000000 <00000012 00000000 <00000012 00000000 <00000012 00000000 <00000012 00000000 <00000012 00000000 <00000014 00000000 <00000014 00000000 <00000014 00000000 <00000014 00000000 <00000014 00000000 <00000014 00000000 <00000004 00000000 <00000004 00000000 <00000004 00000000 <00000004 000000000 <00000004 000000000 <00000004 000000000 <00000004 000000000 <00000001 000000000 <00000001 000000000 <00000001 000000000 <00000001 000000000 <00000001 </th <th></th> <th></th> <th><x00000000c< th=""><th>range : $fffff80 = 128$ Byte</th></x00000000c<></th>			<x00000000c< th=""><th>range : $fffff80 = 128$ Byte</th></x00000000c<>	range : $fffff80 = 128$ Byte
00000000 < < 00000000 < < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 00000000 < 000000000 < 000000000 <		00000000	<x00000014< th=""><th>Tange : TITTTTOO - 120 byce</th></x00000014<>	Tange : TITTTTOO - 120 byce
00000000 <<0000000 00000000 <00000028 00000000 <00000028 00000000 <00000038 00000000 <00000034 00000000 <00000034 00000000 <00000034 00000000 <00000034 00000000 <00000034 00000000 <00000042 00000000 <00000042 00000000 <00000042 00000000 <00000042 00000000 <00000042 00000000 <00000042 00000000 <00000042 00000000 <00000042 00000000 <00000042 00000000 <00000042 00000000 <00000064 00000000 <00000064 00000000 <00000004 00000000 <00000004 00000000 <00000004 00000000 <00000004 00000000 <00000004 00000000 <00000004 00000000 <00000004 00000000 <00000004 00000000 <00000004		00000000	≺x00000018	
00000000 <x00000024< td=""> 00000000 <x00000024< td=""> 00000000 <x00000024< td=""> 00000000 <x00000030< td=""> 00000000 <x00000030< td=""> 00000000 <x00000030< td=""> 00000000 <x00000040< td=""> 00000000 <x00000064< td=""> 00000000 <x00000004< td=""> 00000000 <x00000001< td=""> 00000000</x00000001<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000004<></x00000064<></x00000064<></x00000064<></x00000064<></x00000064<></x00000064<></x00000064<></x00000064<></x00000064<></x00000040<></x00000040<></x00000040<></x00000040<></x00000040<></x00000040<></x00000040<></x00000030<></x00000030<></x00000030<></x00000024<></x00000024<></x00000024<>		00000000	<x000001c< th=""><th>edit memory :</th></x000001c<>	edit memory :
00000000 *X00000024 00000000 *X00000030 00000000 *X00000030 00000000 *X00000030 00000000 *X00000030 00000000 *X00000044 00000000 *X00000064 00000000 *X00000014 00000000 *X00000014 00000000 *X00000014 00000000		00000000	<x00000020< th=""><th></th></x00000020<>	
00000000 -x00000000 00000000 x00000003 00000000 x00000034 00000000 x00000034 00000000 x00000034 00000000 x00000044 00000000 x00000004 00000000 x00000004 00000000 x00000004 00000000 x00000014 00000000 x00000024 00000000 x00000024 00000000 x00000024 00000000 x00000024		00000000	<x00000024< th=""><th></th></x00000024<>	
00000000 x00000030 00000000 x00000030 00000000 x00000030 00000000 x00000044 00000000 x00000064 00000000 x00000066 00000000 x00000066 00000000 x00000064 00000000 x00000064 00000000 x00000006 00000000 x00000006 00000000 x000000014 00000000 x000000024 00000000 x000000024 00000000 x00000024 00000000 x00000024 00000000 x00000026 00000000 x00000026 00000000 x00000026		00000000	<x00000028< th=""><th>Data:</th></x00000028<>	Data:
00000000 ***00000034 00000000 ***00000038 00000000 ***00000040 00000000 ***00000040 00000000 ***00000040 00000000 ***00000040 00000000 ***00000050 00000000 ***00000050 00000000 ***00000050 00000000 ***00000050 00000000 ***00000050 00000000 ***00000050 00000000 ***00000050 00000000 ************************************		00000000	<x00000030< th=""><th>Write Memory View:</th></x00000030<>	Write Memory View:
00000000 *x00000038C 00000000 *x00000038C 00000000 *x00000044 00000000 *x00000044 00000000 *x00000044 00000000 *x00000044 00000000 *x00000054 00000000 *x00000054 00000000 *x00000054 00000000 *x00000056 00000000 *x00000056 00000000 *x00000068 00000000 *x00000068 00000000 *x00000068 00000000 *x00000070 00000000 *x00000006 00000000 *x000000070 00000000 *x00000000 00000000 *x00000000 00000000 *x00000000 00000000 *x000000010 00000000 *x00000010 00000000 *x00000010 000000000 *x00000010 000000000 *x00000010 000000000 *x00000010 000000000 *x00000010 000000000 *x00000010 000000000 *x000000014 000		00000000	≺x00000034	write hemory count
00000000 <x00000040< td=""> 00000000 <x00000044< td=""> 00000000 <x00000050< td=""> 00000000 <x00000000< td=""> 00000000 <x000000000< td=""> 00000000</x000000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000050<></x00000050<></x00000050<></x00000050<></x00000050<></x00000050<></x00000050<></x00000044<></x00000044<></x00000044<></x00000044<></x00000044<></x00000044<></x00000040<>		00000000	≺x0000038	loop on/off
00000000 *x00000044 00000000 *x00000044 00000000 *x00000064 00000000 *x00000064 00000000 *x00000066 00000000 *x00000070 00000000 *x00000070 00000000 *x00000004 00000000 *x00000004 00000000 *x00000004 00000000 *x00000014 00000000		00000000	<x000003c< th=""><th>🗌 refresh view after write 🛛 💆</th></x000003c<>	🗌 refresh view after write 🛛 💆
00000000 -x0000004C 00000000 x0000004C 00000000 x0000005C 00000000 x0000000C 00000000 x00000000 00000000 x000000010 00000000 x000000010 00000000 x000000010 00000000 x000000010 00000000 x000000010 00000000 x00000010 00000000 x000000010		00000000	<x00000040< th=""><th></th></x00000040<>	
00000000 x00000050 00000000 x00000058 00000000 x00000058 00000000 x00000058 00000000 x00000068 00000000 x00000068 00000000 x00000068 00000000 x00000068 00000000 x00000068 00000000 x00000070 00000000 x00000070 00000000 x00000070 00000000 x00000070 00000000 x000000070 00000000 x00000000 00000000 x00000000 00000000 x000000010 00000000 x000000000 00000000 x000000000 00000000 x000000000			<x00000044 <x00000048< th=""><th>mem copy:</th></x00000048<></x00000044 	mem copy:
00000000 *x00000054 00000000 *x00000055 00000000 *x00000055 00000000 *x00000064 00000000 *x00000065 00000000 *x00000006 00000000 *x00000006 00000000 *x00000006 00000000 *x00000006 00000000 *x00000006 00000000 *x000000010 00000000 *x00000010 00000000 *x00000010 00000000 *x00000010 00000000 *x00000010 00000000 *x00000010 00000000 *x00000010 00000000 *x00000012 00000000 *x00000012 00000000 *x00000012 00000000 *x00000014 00000000 *x00000015		00000000	<x0000004c< th=""><th>source</th></x0000004c<>	source
00000000 <x00000058< td=""> 00000000 <x00000058< td=""> 00000000 <x00000068< td=""> 00000000 <x00000006< td=""> 00000000 <x00000006< td=""> 00000000 <x00000006< td=""> 00000000 <x00000008< td=""> 00000000 <x00000008< td=""> 00000000 <x00000014< td=""> 00000000 <x00000014< td=""> 00000000 <x00000012< td=""> 00000000 <x00000014< td=""> 00000000</x00000014<></x00000012<></x00000012<></x00000012<></x00000012<></x00000012<></x00000012<></x00000012<></x00000012<></x00000012<></x00000012<></x00000012<></x00000012<></x00000012<></x00000014<></x00000014<></x00000008<></x00000008<></x00000006<></x00000006<></x00000006<></x00000068<></x00000068<></x00000068<></x00000068<></x00000068<></x00000068<></x00000058<></x00000058<>		00000000	≺x00000050	
00000000 <x0000005c< td=""> 00000000 <x00000064< td=""> 00000000 <x00000062< td=""> 00000000 <x00000062< td=""> 00000000 <x00000062< td=""> 00000000 <x00000062< td=""> 00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x0000004< td=""> 00000000 <x0000004< td=""> 00000000 <x0000004< td=""> 00000000 <x0000004< td=""> 00000000 <x00000010< td=""> 00000000 <x00000012< td=""> 00000000</x00000012<></x00000012<></x00000012<></x00000012<></x00000012<></x00000012<></x00000012<></x00000012<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x0000004<></x0000004<></x0000004<></x0000004<></x00000004<></x00000004<></x00000062<></x00000062<></x00000062<></x00000062<></x00000064<></x0000005c<>		00000000	<x00000054< th=""><th>destination mem copy</th></x00000054<>	destination mem copy
00000000 <x00000060< td=""> 00000000 <x0000064< td=""> 00000000 <x0000066< td=""> 00000000 <x00000060< td=""> 00000000 <x00000070< td=""> 1 Image: 00000000 <x00000004< td=""> 00000000 <x0000000< td=""> 00000000 <x00000000< td=""> 00000000 <x00000000< td=""> 00000000 <x00000010< td=""> <</x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000000<></x00000000<></x0000000<></x0000000<></x0000000<></x0000000<></x0000000<></x00000004<></x00000070<></x00000060<></x0000066<></x0000064<></x00000060<>		00000000	<x00000058< th=""><th></th></x00000058<>	
00000000 <x00000064< td=""> 00000000 <x00000062< td=""> 00000000 <x00000062< td=""> 00000000 <x00000000< td=""> 00000000 <x00000010< td=""> 00000000 <x000000010< td=""> 00000000</x000000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000062<></x00000062<></x00000064<>		00000000	<x0000005c< th=""><th></th></x0000005c<>	
000000000 <x00000066 00000000 <x000000070< td=""> 00000000 <x00000000< td=""> <x00000000< td=""> 00000000 <x00000000< td=""> <x00000000< td=""> 00000000 <x00000000< td=""> 0K 00000000 <x00000014< td=""> 0K 00000000 <x00000014< td=""> 0K 00000000 <x00000014< td=""> 0K 00000000 <x00000014< td=""> 0K 00000000 <x00000012< td=""> 0K 00000000 <x00000012< td=""> 0K 00000000 <x00000014< td=""> 0K 00000000 <x00000012< td=""> 0K 00000000 <x00000012< td=""> 0K 00000000 <x00000014< td=""> 0K <th></th><th></th><th><x00000060< th=""><th>select view range:</th></x00000060<></th></x00000014<></x00000014<></x00000014<></x00000014<></x00000014<></x00000012<></x00000012<></x00000014<></x00000012<></x00000012<></x00000014<></x00000014<></x00000014<></x00000014<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x000000070<></x00000066 			<x00000060< th=""><th>select view range:</th></x00000060<>	select view range:
00000000 <x00000000< th=""> 00000000 <x00000000< td=""> 00000000 <x00000010< td=""> 00000000 <x00000000< td=""> 00000000 <x00000000< td=""> 00000000 <x000000000< td=""></x000000000<></x00000000<></x00000000<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<>		00000000	<x00000068< th=""><th>KB range $(0 - 0) = 0$</th></x00000068<>	KB range $(0 - 0) = 0$
000000000 <00000000 000000000 <00000000 000000000 <00000000 000000000 <00000000 000000000 <00000000 000000000 <00000000 000000000 <00000000 000000000 <00000000 000000000 <00000000 000000000 <00000000 000000000 <00000000 000000000 <00000000 000000000 <00000000 000000000 <00000000 00000000 <00000000 00000000 <00000000 00000000 <00000000 00000000 <00000000 00000000 <00000000 00000000 <00000000 00000000 <00000000 00000000 <00000000 00000000 <00000000 00000000 <00000000 00000000 <00000000 00000000 <00000000 00000000 <00000000 00000000 <00000000		00000000	≺x000006C	
BAR space I 00000000 <x0000000< td=""> 00000000 <x0000000< td=""> 00000000 <x0000000< td=""> 00000000 <x0000000< td=""> 00000000 <x00000000< td=""> 00000000 <x00000000< td=""> 00000000 <x00000000< td=""> 00000000 <x00000010< td=""> 00000000 <x00000000< td=""></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000000<></x00000000<></x00000000<></x0000000<></x0000000<></x0000000<></x0000000<>		00000000	≺x00000070	
BAR space Image: State 00000000 <x00000000< td=""> 00000000 <x00000004< td=""> 00000000 <x00000000< td=""> 00000000 <x000000000< td=""> 00000000 <x000000000< td=""> <</x000000000<></x000000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000000<></x00000004<></x00000000<>			·	
00000000 <x00000000< td=""> 00000000 <x00000004< td=""> 00000000 <x00000000< td=""> 00000000 <x00000000< td=""> 00000000 <x00000000< td=""> 00000000 <x00000010< td=""> 00000000 <x000000010< td=""> 00000000 <x000000014< td=""> 00000000 <x000000054< td=""> 00000000 <x000000064< td<="" th=""><th>1</th><th>⁹</th><th></th><th></th></x000000064<></x000000054<></x000000014<></x000000014<></x000000014<></x000000014<></x000000014<></x000000010<></x000000010<></x000000010<></x000000010<></x000000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000000<></x00000000<></x00000000<></x00000004<></x00000000<>	1	⁹		
00000000 <x00000004< td=""> 00000000 <x00000004< td=""> 00000000 <x00000000< td=""> 00000000 <x00000000< td=""> 00000000 <x00000010< td=""> 00000000 <x00000020< td=""> 00000000 <x00000030< td=""> 00000000 <x00000030< td=""> 00000000 <x00000030< td=""> 00000000 <x00000030< td=""> 00000000 <x00000044< td=""> 00000000 <x00000054< td=""> 00000000 <x00000054< td=""> 00000000 <x00000054< td=""> 00000000 <x00000064< td=""> 00000000 <x00000064< td=""> 00000000 <x00000064< td=""> 00000000 <x00000066< td=""></x00000066<></x00000064<></x00000064<></x00000064<></x00000054<></x00000054<></x00000054<></x00000044<></x00000030<></x00000030<></x00000030<></x00000030<></x00000020<></x00000020<></x00000020<></x00000020<></x00000020<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000010<></x00000000<></x00000000<></x00000004<></x00000004<>		BAR space		
00000000 <x00000004< td=""> 00000000 <x00000000< td=""> 00000000 <x00000014< td=""> 00000000 <x00000014< td=""> 00000000 <x00000012< td=""> 00000000 <x00000012< td=""> 00000000 <x00000012< td=""> 00000000 <x00000024< td=""> 00000000 <x00000024< td=""> 00000000 <x00000024< td=""> 00000000 <x00000024< td=""> 00000000 <x00000034< td=""> 00000000 <x00000034< td=""> 00000000 <x00000034< td=""> 00000000 <x00000044< td=""> 00000000 <x00000044< td=""> 00000000 <x00000050< td=""> 00000000 <x00000054< td=""> 00000000 <x00000056< td=""> 00000000 <x00000064< td=""> 00000000 <x00000064< td=""> 00000000 <x00000065< td=""></x00000065<></x00000064<></x00000064<></x00000056<></x00000054<></x00000054<></x00000054<></x00000054<></x00000054<></x00000054<></x00000050<></x00000044<></x00000044<></x00000034<></x00000034<></x00000034<></x00000024<></x00000024<></x00000024<></x00000024<></x00000012<></x00000012<></x00000012<></x00000014<></x00000014<></x00000000<></x00000004<>		BAR space		
00000000 <x00000000< td=""> 000000000 <x00000010< td=""> 00000000 <x00000010< td=""> 00000000 <x00000012< td=""> 00000000 <x00000012< td=""> 00000000 <x00000024< td=""> 00000000 <x00000026< td=""> 00000000 <x00000030< td=""> 00000000 <x00000030< td=""> 00000000 <x00000034< td=""> 00000000 <x00000044< td=""> 00000000 <x00000044< td=""> 00000000 <x00000050< td=""> 00000000 <x00000050< td=""> 00000000 <x00000050< td=""> 00000000 <x00000050< td=""> 00000000 <x00000054< td=""> 00000000 <x00000056< td=""> 00000000 <x00000064< td=""> 00000000</x00000064<></x00000064<></x00000064<></x00000064<></x00000064<></x00000064<></x00000064<></x00000056<></x00000054<></x00000050<></x00000050<></x00000050<></x00000050<></x00000044<></x00000044<></x00000034<></x00000030<></x00000030<></x00000026<></x00000024<></x00000024<></x00000024<></x00000024<></x00000024<></x00000012<></x00000012<></x00000010<></x00000010<></x00000000<>		00000000	<x0000000< th=""><th>auto read memory OK</th></x0000000<>	auto read memory OK
00000000 <x00000000< td=""> 000000000 <x00000014< td=""> 00000000 <x00000012< td=""> 00000000 <x00000012< td=""> 00000000 <x00000020< td=""> 00000000 <x00000024< td=""> 00000000 <x00000022< td=""> 00000000 <x00000022< td=""> 00000000 <x00000020< td=""> 00000000 <x00000020< td=""> 00000000 <x00000022< td=""> 00000000 <x00000030< td=""> 00000000 <x00000030< td=""> 00000000 <x00000030< td=""> 00000000 <x00000030< td=""> 00000000 <x00000044< td=""> 00000000 <x00000044< td=""> 00000000 <x00000050< td=""> 00000000 <x00000050< td=""> 00000000 <x00000050< td=""> 00000000 <x00000052< td=""> 00000000 <x000000052< td=""> <th>-</th><th>00000000 00000000</th><th><x00000000 <x00000004< th=""><th>auto read memory OK</th></x00000004<></x00000000 </th></x000000052<></x00000052<></x00000052<></x00000052<></x00000052<></x00000052<></x00000052<></x00000052<></x00000050<></x00000050<></x00000050<></x00000044<></x00000044<></x00000030<></x00000030<></x00000030<></x00000030<></x00000022<></x00000020<></x00000020<></x00000022<></x00000022<></x00000024<></x00000020<></x00000012<></x00000012<></x00000014<></x00000000<>	-	00000000 00000000	<x00000000 <x00000004< th=""><th>auto read memory OK</th></x00000004<></x00000000 	auto read memory OK
00000000 <x00000014< td=""> 00000000 <x00000018< td=""> 00000000 <x00000012< td=""> 00000000 <x00000024< td=""> 00000000 <x00000034< td=""> 00000000 <x00000034< td=""> 00000000 <x00000044< td=""> 00000000 <x00000044< td=""> 00000000 <x00000044< td=""> 00000000 <x00000044< td=""> 00000000 <x00000050< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""></x00000060<></x00000060<></x00000060<></x00000050<></x00000050<></x00000050<></x00000050<></x00000050<></x00000050<></x00000050<></x00000050<></x00000050<></x00000044<></x00000044<></x00000044<></x00000044<></x00000034<></x00000034<></x00000024<></x00000024<></x00000024<></x00000024<></x00000024<></x00000024<></x00000012<></x00000018<></x00000014<>		00000000 00000000 00000000	<x00000000 <x00000004 <x00000008< th=""><th>IO Space</th></x00000008<></x00000004 </x00000000 	IO Space
00000000 <x00000018< td=""> 00000000 <x0000001c< td=""> 00000000 <x00000020< td=""> 00000000 <x00000020< td=""> 00000000 <x00000028< td=""> 00000000 <x00000020< td=""> 00000000 <x00000020< td=""> 00000000 <x00000020< td=""> 00000000 <x00000020< td=""> 00000000 <x00000030< td=""> 00000000 <x00000030< td=""> 00000000 <x00000030< td=""> 00000000 <x00000032< td=""> 00000000 <x00000032< td=""> 00000000 <x00000032< td=""> 00000000 <x00000040< td=""> 00000000 <x00000042< td=""> 00000000 <x00000050< td=""> 00000000 <x00000050< td=""> 00000000 <x00000050< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000062< td=""></x00000062<></x00000060<></x00000060<></x00000050<></x00000050<></x00000050<></x00000042<></x00000042<></x00000042<></x00000042<></x00000042<></x00000040<></x00000032<></x00000032<></x00000032<></x00000030<></x00000030<></x00000030<></x00000020<></x00000020<></x00000020<></x00000020<></x00000028<></x00000020<></x00000020<></x0000001c<></x00000018<>		00000000 00000000 00000000 00000000	<x00000000 <x00000004 <x00000008 <x0000000c< th=""><th>IO Space base : 00003000</th></x0000000c<></x00000008 </x00000004 </x00000000 	IO Space base : 00003000
00000000 <x0000001c< td=""> 00000000 <x00000020< td=""> 00000000 <x00000020< td=""> 00000000 <x00000020< td=""> 00000000 <x0000002c< td=""> 00000000 <x0000002c< td=""> 00000000 <x00000030< td=""> 00000000 <x00000030< td=""> 00000000 <x0000003c< td=""> 00000000 <x0000003c< td=""> 00000000 <x00000040< td=""> 00000000 <x00000040< td=""> 00000000 <x00000048< td=""> 00000000 <x00000048< td=""> 00000000 <x00000050< td=""> 00000000 <x00000050< td=""> 00000000 <x00000058< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000068< td=""> 00000000 <x00000068< td=""> 00000000 <x00000062< td=""></x00000062<></x00000068<></x00000068<></x00000060<></x00000060<></x00000060<></x00000060<></x00000058<></x00000050<></x00000050<></x00000048<></x00000048<></x00000040<></x00000040<></x0000003c<></x0000003c<></x00000030<></x00000030<></x0000002c<></x0000002c<></x00000020<></x00000020<></x00000020<></x0000001c<>		00000000 00000000 00000000 00000000 0000	<x00000000 <x00000004 <x00000008 <x00000000 <x00000000< th=""><th>IO Space base : 00003000 range : ffffff80 = 128 Byte</th></x00000000<></x00000000 </x00000008 </x00000004 </x00000000 	IO Space base : 00003000 range : ffffff80 = 128 Byte
00000000 <x000000120< td=""> 00000000 <x00000024< td=""> 00000000 <x00000024< td=""> 00000000 <x00000026< td=""> 00000000 <x00000030< td=""> 00000000 <x00000034< td=""> 00000000 <x00000044< td=""> 00000000 <x00000042< td=""> 00000000 <x00000042< td=""> 00000000 <x00000042< td=""> 00000000 <x00000042< td=""> 00000000 <x00000050< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000062< td=""></x00000062<></x00000060<></x00000060<></x00000060<></x00000050<></x00000050<></x00000050<></x00000050<></x00000050<></x00000050<></x00000050<></x00000050<></x00000042<></x00000042<></x00000042<></x00000042<></x00000044<></x00000034<></x00000034<></x00000034<></x00000034<></x00000034<></x00000030<></x00000026<></x00000024<></x00000024<></x000000120<>		00000000 00000000 00000000 00000000 0000	<x00000000 <x0000004 <x0000008 <x0000000 <x00000010 <x00000014< th=""><th>IO Space base : 00003000 range : ffffff80 = 128 Byte</th></x00000014<></x00000010 </x0000000 </x0000008 </x0000004 </x00000000 	IO Space base : 00003000 range : ffffff80 = 128 Byte
00000000 <x00000024< td=""> 00000000 <x00000028< td=""> 00000000 <x00000028< td=""> 00000000 <x00000034< td=""> 00000000 <x00000036< td=""> 00000000 <x00000044< td=""> 00000000 <x00000048< td=""> 00000000 <x00000054< td=""> 00000000 <x00000058< td=""> 00000000 <x00000060< td=""> 00000000 <x00000068< td=""> 00000000 <x00000068< td=""> 00000000 <x0000068< td=""> 00000000 <x0000068< td=""></x0000068<></x0000068<></x00000068<></x00000068<></x00000060<></x00000058<></x00000054<></x00000048<></x00000044<></x00000036<></x00000034<></x00000034<></x00000034<></x00000034<></x00000034<></x00000028<></x00000028<></x00000024<>		00000000 00000000 00000000 00000000 0000	<pre><x00000000 <x0000004 <x0000008 <x00000000 <x00000010 <x00000014 <x00000018 <x00000016< pre=""></x00000016<></x00000018 </x00000014 </x00000010 </x00000000 </x0000008 </x0000004 </x00000000 </pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte
00000000 <x00000028< td=""> 00000000 <x0000002c< td=""> 00000000 <x00000034< td=""> 00000000 <x00000034< td=""> 00000000 <x00000038< td=""> 00000000 <x00000044< td=""> 00000000 <x00000048< td=""> 00000000 <x00000048< td=""> 00000000 <x00000048< td=""> 00000000 <x00000048< td=""> 00000000 <x00000058< td=""> 00000000 <x00000058< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000068< td=""> 00000000 <x00000068< td=""> 00000000 <x00000068< td=""></x00000068<></x00000068<></x00000068<></x00000060<></x00000060<></x00000058<></x00000058<></x00000048<></x00000048<></x00000048<></x00000048<></x00000044<></x00000038<></x00000034<></x00000034<></x0000002c<></x00000028<>		00000000 00000000 00000000 00000000 0000	<pre><x00000000 <x00000004 <x00000008 <x00000000 <x00000010 <x00000014 <x00000018 <x00000012 <x00000012< pre=""></x00000012<></x00000012 </x00000018 </x00000014 </x00000010 </x00000000 </x00000008 </x00000004 </x00000000 </pre>	IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory :
00000000 <x0000002c< td=""> 00000000 <x00000030< td=""> 00000000 <x00000038< td=""> 00000000 <x00000038< td=""> 00000000 <x0000003c< td=""> 00000000 <x00000040< td=""> 00000000 <x00000040< td=""> 00000000 <x00000048< td=""> 00000000 <x00000048< td=""> 00000000 <x00000048< td=""> 00000000 <x00000050< td=""> 00000000 <x00000050< td=""> 00000000 <x00000058< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000068< td=""> 00000000 <x00000068< td=""> 00000000 <x00000062< td=""></x00000062<></x00000068<></x00000068<></x00000060<></x00000060<></x00000060<></x00000058<></x00000050<></x00000050<></x00000048<></x00000048<></x00000048<></x00000040<></x00000040<></x0000003c<></x00000038<></x00000038<></x00000030<></x0000002c<>		BAR space	<pre><x00000000 <x0000004 <x00000000 <x00000000 <x00000014 <x00000014 <x00000012 <x00000012 <x00000024< pre=""></x00000024<></x00000012 </x00000012 </x00000014 </x00000014 </x00000000 </x00000000 </x0000004 </x00000000 </pre>	edit memory : x45000000 x4500000 x4500000 x4500000 x4500000 x4500000 x4500000 x4500000 x4500000 x4500000 x4500000 x4500000 x450000 x4500000 x45000000 x4500000 x450000 x4500000 x45000000 x45000000 x45000000 x450000000 x45000000 x45000000 x45000000 x45000000 x4500000000 x450000000 x45000000 x45000000 x450000000 x4500000000 x450000000 x4500000000 x450000000 x45000000000 x4500000000000 x45000000000 x450000000000 x450000000000 x45000000000 x450000000000 x450000000000 x4500000000000000 x4500000000000000000 x45000000000000000000000000000000000000
00000000 <x00000030< td=""> 00000000 <x00000030< td=""> 00000000 <x00000038< td=""> 00000000 <x00000030< td=""> 00000000 <x00000030< td=""> 00000000 <x00000040< td=""> 00000000 <x00000040< td=""> 00000000 <x00000048< td=""> 00000000 <x00000048< td=""> 00000000 <x00000048< td=""> 00000000 <x00000050< td=""> 00000000 <x00000058< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000068< td=""> 00000000 <x00000068< td=""> 00000000 <x00000062< td=""></x00000062<></x00000068<></x00000068<></x00000060<></x00000060<></x00000060<></x00000060<></x00000058<></x00000050<></x00000048<></x00000048<></x00000048<></x00000040<></x00000040<></x00000030<></x00000030<></x00000038<></x00000030<></x00000030<>		BAR space	<pre><x00000000 <x0000004 <x00000008 <x00000000 <x00000010 <x00000014 <x00000018 <x00000010 <x00000020 <x00000024 <x00000028< pre=""></x00000028<></x00000024 </x00000020 </x00000010 </x00000018 </x00000014 </x00000010 </x00000000 </x00000008 </x0000004 </x00000000 </pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : x45000000 x45000000
00000000 <x00000034< td=""> 00000000 <x00000034< td=""> 00000000 <x00000036< td=""> 00000000 <x00000040< td=""> 00000000 <x00000044< td=""> 00000000 <x00000048< td=""> 00000000 <x00000048< td=""> 00000000 <x00000054< td=""> 00000000 <x00000056< td=""> 00000000 <x00000068< td=""> 00000000 <x00000062< td=""></x00000062<></x00000068<></x00000068<></x00000068<></x00000068<></x00000068<></x00000068<></x00000056<></x00000054<></x00000048<></x00000048<></x00000044<></x00000040<></x00000036<></x00000034<></x00000034<>		BAR space	<pre><x00000000 x0000004 x00000008 x00000000 <x00000010 <x0000014 x00000018 x00000010 x00000020 x00000024 x00000028 x00000020</x0000014 </x00000010 </x00000000 </pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : x45b00000 < x00003010 1 dwords Data: Tefr.
00000000 <x00000038< td=""> 00000000 <x00000038< td=""> 00000000 <x00000040< td=""> 00000000 <x00000044< td=""> 00000000 <x00000048< td=""> 00000000 <x00000048< td=""> 00000000 <x00000042< td=""> 00000000 <x00000054< td=""> 00000000 <x00000052< td=""> 00000000 <x00000052< td=""> 00000000 <x00000064< td=""> 00000000 <x00000064< td=""> 00000000 <x00000068< td=""> 00000000 <x00000068< td=""> 00000000 <x00000062< td=""></x00000062<></x00000068<></x00000068<></x00000064<></x00000064<></x00000052<></x00000052<></x00000054<></x00000042<></x00000048<></x00000048<></x00000044<></x00000040<></x00000038<></x00000038<>		BAR space	<pre><x00000000 x0000004 <x00000008 <x00000000 <x00000014 <x00000018 <x00000010 <x00000020 <x00000024 <x00000024 <x00000028 <x00000028 <x00000020< pre=""></x00000020<></x00000028 </x00000028 </x00000024 </x00000024 </x00000020 </x00000010 </x00000018 </x00000014 </x00000000 </x00000008 </x00000000 </pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : x45b00000 x00003010 1 dwords Data: Tefr. Write Memory
00000000 <x0000003c< td=""> 00000000 <x00000040< td=""> 00000000 <x00000044< td=""> 00000000 <x00000048< td=""> 00000000 <x00000042< td=""> 00000000 <x00000054< td=""> 00000000 <x00000058< td=""> 00000000 <x00000058< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000068< td=""> 00000000 <x00000068< td=""> 00000000 <x00000062< td=""></x00000062<></x00000068<></x00000068<></x00000060<></x00000060<></x00000060<></x00000058<></x00000058<></x00000054<></x00000042<></x00000048<></x00000044<></x00000040<></x0000003c<>		BAR space	<pre></pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : x4500000 Vrite Memory Write Memory
00000000 <x00000040< td=""> 00000000 <x00000044< td=""> 00000000 <x00000048< td=""> 00000000 <x00000042< td=""> 00000000 <x00000050< td=""> 00000000 <x00000050< td=""> 00000000 <x00000058< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000068< td=""> 00000000 <x00000068< td=""> 00000000 <x00000068< td=""> 00000000 <x00000062< td=""></x00000062<></x00000068<></x00000068<></x00000068<></x00000060<></x00000060<></x00000058<></x00000050<></x00000050<></x00000042<></x00000048<></x00000044<></x00000040<>		BAR space	<pre></pre>	edit memory : x4500000 Write Memory Uoperate in the image is a constructed of the image is a co
00000000 <x00000044< td=""> 00000000 <x00000048< td=""> 00000000 <x00000042< td=""> 00000000 <x00000042< td=""> 00000000 <x00000050< td=""> 00000000 <x00000058< td=""> 00000000 <x00000052< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000068< td=""> 00000000 <x00000068< td=""> 00000000 <x00000062< td=""></x00000062<></x00000068<></x00000068<></x00000060<></x00000060<></x00000052<></x00000058<></x00000050<></x00000042<></x00000042<></x00000048<></x00000044<>		BAR space	<pre><x00000000 <x0000004 <x00000000 <x00000000 <x00000014 <x00000012 <x00000012 <x00000020 <x00000024 <x00000028 <x00000022 <x00000030 <x00000034 <x00000038 <x00000038< pre=""></x00000038<></x00000038 </x00000034 </x00000030 </x00000022 </x00000028 </x00000024 </x00000020 </x00000012 </x00000012 </x00000014 </x00000000 </x00000000 </x0000004 </x00000000 </pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : x45000000 <x00003010 1="" dwords<br="">Data: toggle count view: view: view: view: refr. view: view</x00003010>
00000000 <x00000048< th=""> 00000000 <x0000004c< td=""> 00000000 <x00000050< td=""> 00000000 <x00000054< td=""> 00000000 <x00000058< td=""> 00000000 <x0000005c< td=""> 00000000 <x00000064< td=""> 00000000 <x00000064< td=""> 00000000 <x00000068< td=""> 00000000 <x00000068< td=""> 00000000 <x00000062< td=""></x00000062<></x00000068<></x00000068<></x00000064<></x00000064<></x0000005c<></x00000058<></x00000054<></x00000050<></x0000004c<></x00000048<>		BAR space	<pre></pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : x4500000 <x00003010 dwords<="" l="" td=""> Data: Data: refr. Vrite Memory loop on/off refresh view after write</x00003010>
00000000 <x0000004c< td=""> source 00000000 <x00000050< td=""> destination 00000000 <x00000058< td=""> destination 00000000 <x0000005c< td=""> select view range: 00000000 <x00000068< td=""> KB range (0 - 0): 0</x00000068<></x0000005c<></x00000058<></x00000050<></x0000004c<>		BAR space	<pre></pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : x45b00000 view: view: 0000 on/off refresh view after write
000000000 <x000000054< td=""> 000000000 <x00000054< td=""> 000000000 <x00000058< td=""> 000000000 <x00000050< td=""> 000000000 <x00000060< td=""> 000000000 <x00000064< td=""> 000000000 <x00000068< td=""> 000000000 <x00000068< td=""> 000000000 <x00000068< td=""> 000000000 <x00000062< td=""></x00000062<></x00000068<></x00000068<></x00000068<></x00000064<></x00000060<></x00000050<></x00000058<></x00000054<></x000000054<>		BAR space	<pre></pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : x45b0000 vou0003010 1 dwords Data: toggle view: loop on/off refresh view after write mem copy:
Observed Construction mem copy 00000000 <x00000058< td=""> </x00000058<>		BAR space	<pre></pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : x4500000 x4500000 Vrite Memory Data: refr. view: loop on/off refresh view after write mem copy: source
00000000 <x00000088< td=""> 00000000 <x00000060< td=""> 00000000 <x00000060< td=""> 00000000 <x00000064< td=""> 00000000 <x00000068< td=""> 00000000 <x00000068< td=""> 00000000 <x00000062< td=""></x00000062<></x00000068<></x00000068<></x00000064<></x00000060<></x00000060<></x00000088<>		BAR space	<pre></pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : x4500000 view: view: count view: loop on/off refresh view after write mem copy: source
00000000 <x00000060< td=""> 00000000 <x00000064< td=""> 00000000 <x00000068< td=""> 00000000 <x00000068< td=""> 00000000 <x00000062< td=""></x00000062<></x00000068<></x00000068<></x00000064<></x00000060<>		BAR space	<pre></pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : x4500000 Vrite Memory : 0 toggle refr. view: view: view: view: source destination
00000000 <x00000064< th=""> 00000000 <x00000064< td=""> 00000000 <x00000068< td=""> 00000000 <x00000068< td=""> 00000000 <x00000066< td=""></x00000066<></x00000068<></x00000068<></x00000064<></x00000064<>		BAR space	<pre></pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : x45b00000 <x00003010 dwords<="" l="" td=""> Data: refr. toggle view: loop on/off refresh view after write mem copy: source destination</x00003010>
00000000 <x00000068 (0="" -="" 0):="" 0<="" kb="" range="" th=""><th></th><th>BAR space</th><th><pre></pre></th><th>auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : [x45b0000] <x00003010 1="" dwords<="" td=""> Data: verify loop on/off refresh view after write mem copy: source destination</x00003010></th></x00000068>		BAR space	<pre></pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : [x45b0000] <x00003010 1="" dwords<="" td=""> Data: verify loop on/off refresh view after write mem copy: source destination</x00003010>
00000000 <x0000006c< th=""><th></th><th>BAR space</th><th><pre></pre></th><th>auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : x4500000 volume volume write Memory loop on/off refresh view after write mem copy: source destination mem copy: select view range:</th></x0000006c<>		BAR space	<pre></pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : x4500000 volume volume write Memory loop on/off refresh view after write mem copy: source destination mem copy: select view range:
		BAR space	<pre></pre>	auto read memory OK IO Space base : 00003000 range : ffffff80 = 128 Byte edit memory : x4500000 void to the second
00000000 <x00000070< th=""><th></th><th>BAR space</th><th><pre><x00000000 x0000004 <x0000000 x0000000 x00000010 <x00000010 <x00000010 <x00000020 <x00000020 <x00000020 <x00000020 <x00000020 <x00000030 <x00000030 <x00000034 <x00000038 <x00000034 <x00000044 <x00000044 <x00000044 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x0000000000< th=""><th>auto read memory OK IO Space base : 00003000 range : fffff80 = 128 Byte edit memory : x4500000 Vrite Memory Data: refr. view: count loop on/off refresh view after write mem copy: source destination mem copy: select view range: KB range (0 - 0): 0</th></x0000000000<></x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000044 </x00000044 </x00000044 </x00000034 </x00000038 </x00000034 </x00000030 </x00000030 </x00000020 </x00000020 </x00000020 </x00000020 </x00000020 </x00000010 </x00000010 </x0000000 </x00000000 </pre></th></x00000070<>		BAR space	<pre><x00000000 x0000004 <x0000000 x0000000 x00000010 <x00000010 <x00000010 <x00000020 <x00000020 <x00000020 <x00000020 <x00000020 <x00000030 <x00000030 <x00000034 <x00000038 <x00000034 <x00000044 <x00000044 <x00000044 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x00000050 <x0000000000< th=""><th>auto read memory OK IO Space base : 00003000 range : fffff80 = 128 Byte edit memory : x4500000 Vrite Memory Data: refr. view: count loop on/off refresh view after write mem copy: source destination mem copy: select view range: KB range (0 - 0): 0</th></x0000000000<></x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000050 </x00000044 </x00000044 </x00000044 </x00000034 </x00000038 </x00000034 </x00000030 </x00000030 </x00000020 </x00000020 </x00000020 </x00000020 </x00000020 </x00000010 </x00000010 </x0000000 </x00000000 </pre>	auto read memory OK IO Space base : 00003000 range : fffff80 = 128 Byte edit memory : x4500000 Vrite Memory Data: refr. view: count loop on/off refresh view after write mem copy: source destination mem copy: select view range: KB range (0 - 0): 0



• To view the ISA waveforms for IO write first set the trigger values as in the screenshot & trigger for these values by pressing F5 & then press write memory tab.we can observe the waveform as shown in the IO write screenshot

🕲 ChipScope Pro Analyzer [isa_	bu	s]					
<u>File View J</u> TAG Chain <u>D</u> evice	Tri	gger S	etup W <u>a</u> veform <u>W</u> indow <u>H</u> elp				
# ● I I I I I I I I I I I I I I I I I I							
Project: isa_bus	1	Triaae	er Setup - DEV:1 MvDevice1 (XC3S1000) UI	NIT:0 MVILAO (ILA)		р ^к в ^л 🛛	
JTAG Chain 🔄	<u> </u>		Match I Init	Eunction	Value	Radiy Counter	
- DEV:0 MyDevice0 (XCF08P	Nat	•- M∩:	TriggerPort0	==	X X000X X000X X000X X000	Hex disabled	
- DEV.1 WyDevice1 (XC3S10 - UNIT:0 MyIL 40 (ILA)	÷	-	(app(PIO(bale o			(
- Triager Setup			(app(PIO/bclk_o				
- Waveform -			(app/PIO/mamw_n_a				
					^		
Signals: DEV: 1 UNIT: 0			Japp/PIO/memr_n_o		^	1	
Data Port			/app/PIO/iow_n_o		U		
- /app/PIO/Ia o		-	/app/PIO/ior_n_o		×		
∽ /app/PIO/PIO_EP/ep_isa		-	/app/PIO/PIO_EP/ep_isa/rd_data_o<15>		×		
🗠 /app/PIO/sa_o		-	/app/PIO/PIO_EP/ep_isa/rd_data_o<14>		×		
🗠 /app/PIO/sdout_o		-	/app/PIO/PIO_EP/ep_isa/rd_data_o<13>		×	(
— CH: 0 /app/PIO/sa_o<0>		_	/app/PIO/PIO EP/ep isa/rd data o<12>		×		
- CH: 1 /app/PIO/sa_o<1>		_	/app/PIO/PIO_EP/ep_isa/rd_data_o<11>		×	(
CH: 2 /app/PIO/sa_0<24			(ann/PIO/PIO EP/an jea/rd data or10>		×		
- CH: 4 /app/r10/sa_0<4>			/app/110/110_E1/ep_isa/rd_data_o-10-				
- CH: 5 /app/PIO/sa_o<5>		_	/app/FIO/FIO_EF/ep_isa/ru_uata_0<9>		~		
— CH: 6 /app/PIO/sa_o<6>			/app/PIO/PIO_EP/ep_isa/rd_data_o<8>		7		
— CH: 7 /app/PIO/sa_o<7>			/app/PIO/PIO_EP/ep_isa/rd_data_o<7>		×		
— CH: 8 /app/PIO/sa_o<8>		-	/app/PIO/PIO_EP/ep_isa/rd_data_o<6>		×	(
— CH: 9 /app/PIO/sa_o<9>		-	/app/PIO/PIO_EP/ep_isa/rd_data_o<5>		×	(
- CH: 10 /app/PIO/sa_o<1		-	/app/PIO/PIO_EP/ep_isa/rd_data_o<4>		×		
- CH: 11 Japp/PIO/sa_o<1		_	/app/PIO/PIO EP/ep isa/rd data o<3>		×	(
- CH: 13 (ann/PIO/sa_0<1			(ann/BIO/BIO ER/an ica/rd data ar25			· · · · · · · · · · · · · · · · · · ·	
- CH: 14 /app/PIO/sa_o<1	v [
- CH: 15 /app/PIO/sa_o<1	Ξ	Add	Active Trigger Co	ndition Name	Trigger Condition E	quation	
- CH: 16 /app/PIO/sa_o<1	9	Del	Trigger	Condition0	MO		
CH: 17 /app/PIO/sa_o<1	-	_		- 1			
- CH: 18 /app/PIO/sa_o<1	ap	Type:	Window Windows:	1 Depth:	512 Position:	64	
- CH: 19 /app/PIO/sa_o<1	ture	Otors	a Quelification	All Dat-			
CH: 20/app/PIO/ia_0<1	0	Storag	e quanication:	Ali Data			
INFO - Device 1 Unit 0: Waiting for c	ore	to be a	armed			×	
Upload						DONE	
🛃 start 🛛 🧐 🙆 📔)2\	Window:	s 🔹 🧐 Outlook Expr 🛛 🐔 Adobe Re	ad 🕘 ChipScop	e Pr 🍟 isa2.bmp - P 📓 Calcul	ator 🔇 🔽 🗊 1:40 PM	



• IO write Screenshot



Figure 2: IO Write Cycle



• To read from the IO space of the host cpu,mark the auto read memory tab select some register in the top down list. To check the ISA waveform in the chipscope viewer set the trigger values in the trigger window as shown in screenshot press F5 to trigger and press the refresh view window to read the the register of IO space.

🕲 ChipScope Pro Analyzer [isa_	bı	is]									
<u>File View J</u> TAG Chain <u>D</u> evice	Ţr	gger S	ietup W <u>a</u> veform <u>W</u> in	dow <u>H</u> elp							
□ · · · · · · · · · · · · · · · · · · ·											
Project: isa_bus	0	Trigge	er Setup - DEV:1 MyDev	rice1 (XC3S1000) UN	IT:0 MyILAO	(ILA)				្រ រ	
JTAG Chain 🔺	2		Match	h Unit	e	Radix	Counter				
DEV:0 MyDevice0 (XCF08P	Nato	♀ M0;	:TriagerPort0			==	X_?X00(_)	- 000(_X0X0_1802	Hex	disabled	-
 UNIT:0 MVII A0 (ILA) 	3	· -	/app/PIO/bale o					X			
- Trigger Setup		_	/app/PIO/bclk_o					X			
– Waveform 🔍		_	Jann/PlO/memw_n_n					X			
		_	/app/PIO/memr_n_o					X			
Signals: DEV: 1 UNIT: 0			(ann/PIO/iow_n_o					X			
💡 Data Port 🔄			(app/PIO/ior_n_o								
► /app/PIO/Ia_o			(app/10/PIO/PIO_EP/en_is	<15> etch hries				X			
app/PIO/PIO_EP/ep_isa			(ann/PIO/PIO_EP/ep_ic	<pre>sand_data_oris> </pre>				X			
 /app/PIO/sdout_o 			(ann/PIO/PIO_EP/ep_ic	<pre>sand_data_ort+*</pre>				X			
— CH: 0 /app/PIO/sa_o<0>			/app/10/10_L1/ep_is	sanu_uata_o<13>				X			
— CH: 1 /app/PIO/sa_o≺1>			/app/10/10_E1/ep_ic	sand_data_o<12>				X			
- CH: 2 /app/PIO/sa_o<2>			/app/FIO/FIO_EF/ep_is	sa/iu_uala_0<112				~ ~ ~			-
- CH: 4 /ann/PIO/sa_0<4>			/app/FIO/FIO_EF/ep_is	sanu_uata_o <io></io>				~ ~ ~			-
- CH: 5 /app/PIO/sa_o<5>			/app/PIO/PIO_EP/ep_is	sa/ru_uata_u<9>				~ ~			- 1
— CH: 6 /app/PIO/sa_o≺6>		- (ann/PIO/PIO_EP/ep_isa/rd_data_o<7>						~ ~			- 1
- CH: 7 /app/PIO/sa_o<7>		- /app/PIO/PIO_EP/ep_isa/rd_data_o						~ ~			-
- CH: 8 /app/PIO/sa_o<8×		/app/PIO/PIO_EP/ep_isa/rd_data_o<6>						X			- 1
- CH: 10 /app/PIO/sa_0<9		/app/PIO/PIO_EP/ep_isa/rd_data_o<5>						X			_
- CH: 11 /app/PIO/sa_o<1			/app/PIO/PIO_EP/ep_is	sa/rd_data_o<4>				X			_
— CH: 12 /app/PIO/sa_o<1		-	/app/PIO/PIO_EP/ep_is	sa/rd_data_o<3>				X			
- CH: 13 /app/PIO/sa_o<1			Jonn/DIA/DIA EDian in	nird data ar25				¥			
 CH: 14 Japp/PIO/sa_0<1 CH: 15 Japp/PIO/sa_0<1 	E	Add	Active	Trigger Cor	ndition Name		Tri	igger Condition Ec	quation		
- CH: 16 /app/PIO/sa_o<1	ö	Del	۲	TriggerC	ondition0			MO			-
- CH: 17 /app/PIO/sa_o<1	•										
- CH: 18 /app/PIO/sa_o<1	Cap	Туре:	Window	Windows:	1	Depth: 51	12 💌	Position:		64	
- CH: 19 /app/PIO/sa_o<1	ture	Qtorac	no Qualification:		All	Data					
		otora <u>c</u>	ge Qualification.		00	Data					
INFO - Device 1 Unit 0: Waiting for c	core	to be a	armed								^
Upload										DON	1E
🛃 start 🛛 🟟 🚳 😜 📄) 3	Window	ıs Explorer 🛛 🖌 🧐 Ou	itlook Express - Arun	🕘 ChipSco	pe Pro Analy:	/z 🦉 isa5.bmp - i	Paint	(< V2 🕫 🧶 2:2	28 PM



• IO read screenshot.



Figure 3: IO Read Cycle



• To write into the memory space of the host cpu, select memory BAR register & double click on that BAR register & press yes tab on the information window.

2 PciTree		
direct select: bus: dev 3 ÷ 0	func:	highest show Mem Map 3.0.0 PCI/ISA: Bridge Device
0.00.0 0.28.0 0.28.4 0.28.4	Host/PCI; Bridge I 0->1 (2) PCI/PCI; Br: 1->2 (2) PCI/PCI; 0->3 (3) PCI/PCI; Br: PCI/IS&: Bridge	VID: x10EE Xilinx Corp DID: x0007 no device name found no · SubVID: x10EE Xilinx SubID: x0007 no-name
	0->4 (4) PCI/PCI; Br: Ethernet; Netwo 16550 Compatibl o. serial bus C Universal Host Cor Universal Host Cor Universal Host Cor	edit ConfReg: x48000000 hex Write ConfReg refr after wr.
0.29.3 0.29.7 0.30.0 5.04.0 0.31.0 0.31.1 0.31.3	Universal Host Cor o. serial bus Dev: O->5 (5) Subtractive, VGA; PC Compati Ethernet; Netwo PCI/ISA; Bridge De o. Mass Storage Co SMBus; Serial Bus	Config Space Dump: (type 1 xs) 0007 10EE <00 : DID VID 0010 0147 <04 : Stat Cmd 0601 0000 <08 : BaseClass SubClass I 0000 0010 <0C : BIST Header LatTimer 0000 3001 <10 : BAR 0 io 4800 0000 <14 : BAR 1 mem 32bit 0000 0000 <18 : BAR 2 0000 0000 <1C : BAR 3

Ē	PciTree			
	direct select: bus: dev: 3 5 0	func:	ow INT routing highest B X I T show Mem Map 5 About	
	Host CPU 0.00.0 0.28.0 0.28.4	Host/PCI; Bridge I 0->1 (2) PCI/PCI; Br: 1->2 (2) PCI/PCI;	PCI/ISA; Bridge Device VID: x10EE Xilinx Corp DID: x0007 no device name found no SubVID: x10EE Xilinx	
		0-> Are you ready t BAR read-write If not: a BARrai	to perform e(0xfffffff)-read-restore ?	-
	0.29.1 0.29.2 0.29.3 0.29.7	Yes N o. serial bus Dev:	Cancel Cancel Concel Concel (type 1 xs)] —1
		VGA; PC Compati Ethernet; Netwo PCI/ISA; Bridge De o. Mass Storage Co SMBus; Serial Bus	0010 0147 <04	I



- The window containing all the registers under that BAR space are listed, we can access those register. To write into any register of memory space select some register, enter the data to be written in edit memory tab.
- To view the ISA waveforms for memory write first set the trigger values as in the screenshot & trigger for these values by pressing F5 & then press write memory tab.we can observe the waveform as shown in the memory write screenshot

ChipScope Pro Analyzer [isa_bi	us]								
<u>File View J</u> TAG Chain <u>D</u> evice <u>T</u> r	riggei	r Setup W <u>a</u> veform <u>W</u> ind	ow <u>H</u> elp						
	ę I	₽ ₽ IZ							
Project: isa bus	ан т. Т. т. т.	num Cature DD 64 M David	and (MC2E400E) LIN	T-0 64-41 AO /11 A1 22					· 🖂
JTAG Chain	∦ TLU	gger Setup - DEV:1 MyDeVi	CE1 (XC351000) UN	IT:U MIYILAU (ILA) 🎡					
- DEV:0 MyDevice0 (XCF08P		Match	Unit	Function		Value	Radix	Counter	
e-DEV:1 MyDevice1 (XC3S10 🔤	<u>9</u> 1	MO:TriggerPort0		==	X_	?x00(_x000(_x000(_x000(Hex	disabled	
		– /app/PIO/bale_o				Х			
Trigger Setup		– /app/PIO/bclk_o				Х			
- Waveform		/app/PIO/memw n o				0			=
		- /app/PIO/memr_n_o				Х			
Signals: DEV: 1 UNIT: 0		- (app/PIO/iow_n_o				X			
💡 Data Port 🔄	-	/app/iio/iow_ii_o				×			
🗠 /app/PIO/la_o		- Japp/PiO/lor_n_o				A			- 11
► /app/PIO/PIO_EP/ep_isa		- /app/PIO/PIO_EP/ep_is	a/rd_data_o<15>			Χ			_
► /app/PIO/sa_o		/app/PIO/PIO_EP/ep_is	a/rd_data_o<14>			X			
Japp/PIU/sdout_o OUI: 0 /app/PIO/soc.oc/0		-/app/PIO/PIO_EP/ep_is	a/rd_data_o<13≻			Х			
CH: 1 (app/PIO/sa_oc1>		-/app/PIO/PIO_EP/ep_is:	a/rd_data_o<12>			Х			
- CH: 2 (ann/PIO/sa_0<2>		-/app/PIO/PIO_EP/ep_is;	a/rd data o<11>			Х			
- CH: 3 /app/PIO/sa_o<3>		- (ann(PIO(PIO_EP(en_is)	<pre></pre>			X			
- CH: 4 /app/PIO/sa_o<4>-		/app/10/10_EF/ep_ist	and_data_o <to></to>			v			- 11
— CH: 5 /app/PIO/sa_o<5>		/app/FIO/FIO_EF/ep_Is	a/iu_uala_0<92			~ ~ ~			- 11
— CH: 6 /app/PIO/sa_o<6>		- /app/PIO/PIO_EP/ep_is	a/rd_data_o<8>			Χ			- 11
— CH: 7 /app/PIO/sa_o<7>		– /app/PIO/PIO_EP/ep_is;	a/rd_data_o<7>			X			_
— CH: 8 /app/PIO/sa_o<8>		-/app/PIO/PIO_EP/ep_is	a/rd_data_o<6≻			Х			
— CH: 9 /app/PIO/sa_o<9>		-/app/PIO/PIO_EP/ep_is:	a/rd_data_o<5>			Х			
- CH: 10 /app/PIO/sa_o<1		-/app/PIO/PIO EP/ep is:	a/rd data o≺4>			Х			
- CH: 11 Japp/PIO/sa_o<1 :		- (ann/PIO/PIO_EP/en_is:	eind data o<3>			X			
- CH: 13 (ann/PIO/sa_0<1		74pp/10/10_E17cp_10	and_data_0.5			~			
- CH: 14 /app/PIO/sa_o<1	0 al.	d Botino	Triagor Con	dition Nomo		Triggor Condition F.	nuction		
- CH: 15 /app/PIO/sa_o<1	Add	Active	TriggerCon	andition0		M0	qualion		-
- CH: 16 /app/PIO/sa_o<1	De		mygerc	onulliono		MU			
— CH: 17 /app/PIO/sa_o<1	Tur	a: Window	Alindaula:	4 5	. 510	Desition		64	
– CH: 18 /app/PIO/sa_o<1	Тур	e. window 💌 🔪	winuows:	i Deptr	. 512	Position:		04	
- CH: 19 /app/PIO/sa_o<1	Stor	rage Qualification:		All Data					
- CH: 20 /app/PIO/la_o<11	0.01	rage sudmitation.		All Data					
	- 4- 1-								
INFO - Device 1 Onit U: Waiting for con	e to b	ie anned							*
Upload								DOI	NE
🐉 start 🛛 🟟 🚳 😜 📄 2	Wind	lows Expl 🔹 🧐 Outlook	Express 📓 Cl	hipScope Pro A	Calculator	🦉 isa9.bmp - Pair	nt 🤇	VI 🖉 🌒 2:	38 PM



• Memory write Screenshot

🕲 ChipScope Pro Analyzer [isa]	_bus]											
<u>F</u> ile <u>V</u> iew <u>J</u> TAG Chain <u>D</u> evice	<u>T</u> rigger Setup W <u>a</u> veform <u>W</u> indow <u>H</u> elp											٦
👯 🕑 🕨 🖿 T! 🛛 🛠 😚 🦷	? <u>२</u>											
Project: isa_bus	Waveform - DEV:1 MyDevice1 (XC3S1000)) UNIT	:0 M	/LAO (ILA)							c c 🛛	
JTAG Chain — DEV:0 MyDevice0 (XCF08P	Bus/Signal	x	0	64 -24 Cl ⊢	16 D	56 96	136 17	76 216	256 296 3	336 376 4	116	1
DEV:1 MyDevice1 (XC3S10 DEV:1 MyDevice1 (XC3S10	∽ /app/PIO/PIO EP/ep isa/rd data o	341	341	3412	X			7698			<u>}</u>	
Trigger Setup	← /app/PIO/sdout o	341 :	341	3412				7698				
- Waveform	- ~ /app/PIO/la o	00	00					00			5	
Signals: DEV: 1 UNIT: 0	∽ /app/PIO/sa o	000	000	00010)				000A6				
P-Data Port	- /app/PIO/sdout_en_o	0	0									
► /app/PIO/Ia_o ► /app/PIO/PIO EP/ep isa	- /app/PIO/ior n o	1	1								_	
✓ /app/PIO/sa_o	/app/PIO/iow n o	1	1									
► /app/PIO/sdout_o CH: 0 /app/PIO/sa_o<0>	/ann/PTO/mewr n o	1	1								_	
— CH: 1 /app/PIO/sa_o<1≍	/ann/PTO/memm n o	1	1									
— CH: 2 /app/PIO/sa_o<2> — CH: 3 /app/PIO/sa_o<3>	/app/FIO/mclk_o	1	1	hannan	 חחחחר	חחחחחחחח				ותחתחתחח	0000	
− CH: 4 /app/PIO/sa_o<4≍−	/app/FIO/DOIK_0		1	r	UUUU		JUUUUUUUUU		1000000000000			
— CH: 5 /app/PIO/sa_o<5> — CH: 6 /app/PIO/sa_o<6>	- /app/PlU/bale_0	U	U									
- CH: 7 /app/PIO/sa_o<7>												
– CH: 8 /app/PIO/sa_o<8>												
- CH: 9 /app/PIO/sa_o<9>												
- CH: 11 /app/PIO/sa_o<1												
— CH: 12 /app/PIO/sa_o<1												
- CH: 13 /app/PIO/sa_o<1												
- CH: 15 /app/PIO/sa_o<1												
— CH: 16 /app/PIO/sa_o<1												
- CH: 17 /app/PIO/sa_o<1			1								-	
 CH: 18 /app/PIO/sa_0<1 CH: 19 /app/PIO/sa_0<1 			•								•	
- CH: 20 /app/PIO/Ia_o<1				X: -64		• • 0	-64	4 >	∆(X-0):0			Į
INFO - Device 1 Unit 0: Waiting for a	core to be armed											*
Upload											DONE	
🐉 start 🛛 🟟 🚳 📔	2 Windows Explorer 🔹 🧐 Outlook Express -	Arun		ChipScop	e Pro A	analyz	Calculator			< V2 2 4	🦻 2:38 PM	

Figure 4: Memory Write Cycle



• To read from the memory space of the host cpu,mark the auto read memory tab select some register in the top down list. To check the ISA waveform in the chipscope viewer set the trigger values in the trigger window as shown in screenshot press F5 to trigger and press the refresh view window to read the the register of memory space.

ChipScope Pro Analyzer [isa	a_bi	IS]				X
<u>F</u> ile <u>V</u> iew <u>J</u> TAG Chain <u>D</u> evice	ιI	igger Setup W <u>a</u> veform <u>W</u> indow <u>H</u> elp				
🟥 🕑 🕨 🔳 T! 🛠 🛇	Ŷ	우 / 오 교				
Project: isa_bus		Trigger Setup - DEV:1 MyDevice1 (XC3S1000) UNIT:0 My	/LAO (ILA)		r ا ا ا	
JTAG Chain	2	Match Unit	Function	Value	Radix Counter	
DEV:0 MyDevice0 (XCF08P	lato	P−M0:TriggerPort0	==	X_?X00(_X000(_X000(_X000(_X000(Hex disabled	•
 UNIT:0 MyILAD (ILA) 	7	-/app/PIO/bale_o		X		
- Trigger Setup		- (app/PIO/bclk_o		×		
- Waveform		- (app(PIO/memy, n. o		0		_
				v		
Signals: DEV: 1 UNIT: 0		- /app/FIO/Interni_n_o		^		
👇 Data Port 🛛		- JappiPIOJiow_n_o		X		
🗠 /app/PIO/Ia_o		/app/PIO/ior_n_o		×		
∽ /app/PIO/PIO_EP/ep_isa		/app/PIO/PIO_EP/ep_isa/rd_data_o<15>		X		
⊶ /app/PIO/sa_o		<pre>/app/PIO/PIO_EP/ep_isa/rd_data_o<14></pre>		X		
 App/PIO/sdout_o App/PIO/sdout_o 		-/app/PIO/PIO_EP/ep_isa/rd_data_o<13>		Х		
CH: 0 /app/PIO/sa_0<04		/app/PIO/PIO_EP/ep_isa/rd_data_o<12>		Х		
- CH: 2 (ann/PIO/sa_0<2>		/app/PIO/PIO EP/ep isa/rd data o<11>		X		
- CH: 3 /app/PIO/sa_o<3>		/app/PIO/PIO_EP/ep_isa/rd_data_o<10>		X		
— CH: 4 /app/PIO/sa_o≺4>—				X		
— CH: 5 /app/PIO/sa_o<5>				v		
— CH: 6 /app/PIO/sa_o<6>		Tapp/PIO/PIO_EP/ep_isa/ru_data_0<8>		^		
— CH: 7 /app/PIO/sa_o<7≯		- /app/PIO/PIO_EP/ep_isa/rd_data_o		X		
- CH: 8 /app/PIO/sa_o<8>		/app/PIO/PIO_EP/ep_isa/rd_data_o<6>		X		
CH: 10 (app/PIO/sa_0<94		<pre>/app/PIO/PIO_EP/ep_isa/rd_data_o<5></pre>		X		
- CH: 11 /ann/PIO/sa_o<1		— /app/PIO/PIO_EP/ep_isa/rd_data_o<4>		X		
- CH: 12 /app/PIO/sa_o<1		/app/PIO/PIO_EP/ep_isa/rd_data_o<3>		X		_
- CH: 13 /app/PIO/sa_o<1		ji				
— CH: 14 /app/PIO/sa_o<1		Add Active Trigger Condition	Name	Trigger Condition E	quation	
— CH: 15 /app/PIO/sa_o<1	rig	Dol TriggerConditio	in0	MO		<u>-</u>
- CH: 16 /app/PIO/sa_o<1	Ļ		1			1
- CH: 17 Japp/PIO/sa_o<1	ဂ္ဂ	Type: Window 🔻 Windows:	1 Depth:	512 Position:	64	
- CH: 19 (ann/PIO/sa_0<1	pt		· [
- CH: 20/app//10/3a_o<1	Ē	Storage Qualification:	All Data			
INFO - Device 1 Unit 0: Waiting for	con	e to be armed				*
Upload					DONE	
🐉 start 🛛 🟟 🕲 🚺	2	Windows Expl 🔹 🧐 Outlook Express 🛛 👰 ChipScop	e Pro A	Calculator 🦉 isa9.bmp - Pai	nt 🔇 🔽 🗾 🧐 2:38 F	M



• Memory read screenshot.



Figure 5: Memory Read Cycle



APPENDIX A Reference Documents

- PCItree software usage from <u>http://www.pcitree.de/userguide.html</u>
- Spartan-3 for PCI Express starter kit board user guide UG256 -<u>http://www.xilinx.com/support/documentation/boards_and_kits/ug256.pdf</u>