





1st Mesoamerican Workshop on Reconfigurable X-ray Scientific Instrumentation for Cultural Heritage

The RVI communication block: ComBlock

Antigua Guatemala, June 2025







Maynor Ballina



International Centre for Theoretical Physics

Introduction

- Vendor specific interconnects (AXI, AVALON, Chiplink).
- What we learned yesterday?
 - ✓ AXI interconnect,
 - ✓ memory address,
 - ✓ data.
- Communication Block (ComBlock).





What is the ComBlock?

A communication block that abstracts you from the AXI protocol.

- Interconnects: AXI Known interfaces (registers, RAM and FIFOs).
- Configurable resources, width, length, etc.
- Simple Clock-Domain-Crossing with FIFOs and TDPRAM.
- C driver for easy complete project integration.



Features

Input/Output Interfaces:

- I/O Registers
- I/O FIFOs
- TDPRAM (True Dual Port RAM) Memory

AXI Compatibility:

- AXI Lite: Supports access to registers and FIFOs through this lightweight, low-bandwidth interface.
- AXI Full: Used to access TDPRAM, ideal for larger, high-performance data transfers.



Registers

16 Input Registers: Communication from the FPGA to the processor.16 Output Registers: Communication from the processor to the FPGA.

Flexible Configuration:

- User-configurable data width: From 1 to 32 bits, depending on application requirements.
- Independent enable for input and output registers: Allowing a more optimized design tailored to different communication needs.

			Registers					
∷ + AXIL II IN_REGS	OUT_REGS		Input (FPGA to P	ROC)		Output (PROC	to FPGA)	
 reg0_i[31:0] reg1_i[31:0] 	reg0_o[15:0] 🕨 • reg1_o[15:0] 🕨 •		🕑 Enable			🕑 Enable		
- axil_aclk	reg2_o[15:0] 🕨	•	Data Width 3	2	🔊 [1 - 32]	Data Width	16	🛞 [1 - 32]
)		Quantity 2		🔉 [1 - 16]	Quantity	3	🛞 [1-16]

FIFOs

Input and Output FIFOs: Input: FPGA → Processor, Output: Processor → FPGA

Customizable Width and Depth: Both the data width and FIFO depth can be configured.

Asynchronous Operation: The FIFOs support safe clock domain crossing, enabling integration into heterogeneous systems.

Processor-Side Control Registers: Allow reading of occupancy level, as well as resetting or clearing the queues.

	FIFOs				
	Input (FPGA to PROC)			Output (PROC to FPGA)
→ fifo_data_i(15:0) → fifo_re_i ◄ -	🕑 Enable			🕑 Enable	
- < fifo_full_o fifo_data_o(7:0) -	Data Width	16	[1 - 32]	Data Width	8 (1 - 32)
■ ◀ fifo_overflow_o fifo_empty_o ► = fifo_clk i fifo_aempty_o ► =	Depth	1024	8	Depth	256
fifo_clear_i fifo_underflow_o ▶	Almost Full Offset	1	8	Almost Full Offset	1
	Almost Empty Offset	1	8	Almost Empty Offset	1

TDPRAM

Simultaneous Bidirectional Access: The FPGA and processor can perform read and write operations independently and concurrently.

Clock Domain Crossing (CDC): Designed to operate in systems with different clock domains, enabling safe and lossless data transfer.

User-Configurable Parameters: Data width, address width, and depth.

Example: If the address width is 10 bits, the default depth will be $2^{10} = 1024$.

	+ AXIF	True Dual Port RA	M (FPGA to PROC,	PROC to FPGA)
	- IO_DRAM ram_we_i	🕑 Enable		
_	 ram_addr_i[15:0] ram_data_i[7:0] 	Data Width	8	0 [1 - 32]
-	<pre>arm_data_o[7:0] ram_clk_i ''''''''''''''''''''''''''''''''''''</pre>	Address Width	16	0 [1 - 32]
0	axif_aresetn	Depth	0	3

Processor Interaction

- Device information in **xparameters.h**.
- Register mappings in **comblock.h**.
- CB_IREG, CB_OREG, ...
- Control registers.



C Drives

• Read and write functions **single memory position**:

void **cbWrite**(UINTPTR baseaddr, u32 reg, u32 value) u32 **cbRead**(UINTPTR baseaddr, u32 reg)

Read and write several contiguous memory positions:

void **cbWriteBulk**(UINTPTR baseaddr, int *buffer, u32 depth) void **cbReadBulk**(int *buffer, UINTPTR baseaddr, u32 depth)







1st Mesoamerican Workshop on Reconfigurable X-ray Scientific Instrumentation for Cultural Heritage

Lab 2: Reconfigurable instrumentation on SoC-FPGA

Antigua Guatemala, June 2025









Maynor Ballina



International Centre for Theoretical Physics

Labs 2: Objectives

- •Simulate a cross-level trigger and understand the VHDL code.
- •Use and configuration of the **ComBlock** IP.
- •Understand the data transfer between PL and PS using ComBlock IP.
- Instantiate RTL blocks and control them through the ComBlock

registers.

•Perform data streaming to the **oscilloscope** through the **ComBlock** FIFO.

•Test the complete design on the ZedBoard platform to verify the implementation.

Labs 2: Design description



Labs 2: Oscilloscope



Labs 2: Simulation

Untitled 1 × tb_cross	LevelTrigger.vhd	× ? 🗆 🖸
ର 🔛 ବ୍ ର 🔀	∦ + [)	I 1 12 12 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
		1,000.000 ns
Name	Value	0.000 ns 200.000 ns400.000 ns600.000 ns800.000 ns
🖁 clk	0	a fa
Ъ aresetn	1	
> 😻 din[7:0]	5f	
> 😻 threshold[7:0]	64	64 1
🔓 edgeSel	1	
la trigger	0	
UATA_BUS_WIDTH	8	
	× 2	

Labs 2: Hardware

Comblock Configuration

<u>}</u>	Re-customize IP × X
ComBlock (2.0)	🔺
🚯 Documentation 🛛 🖨 IP Location	
Show disabled ports	Component Name comblock_0 Registers
	Input (FPGA to PROC) Output (PROC to FPGA)
	Enable Enable
	Data Width 8 Image: Second se
	Quantity 2 Image: [1 - 16] Quantity 4 Image: [1 - 16]
+ AXIL + IN_REGS + IN_FIFO fifo_clk_i OUT_REGS + I fifo_clear_i axil_aclk • axil_aresetn	Input (FPGA to PROC) Output (PROC to FPGA) Imput (FPGA to PROC) Imput (PROC to FPGA) Imput (PROC) Imput (PROC to FPGA)
	OK Cancel

15

Labs 2: Hardware



Labs 2: Hardware



Labs 2: Final Block Design



Labs 2: Software

- C driver using ComBlock instructions
- Handles configuration and data exchange.
- Uses the waveform.h file.
- Allows changes to threshold, pretrigger samples, and pulse length.
- Software interface lets you adjust parameter values.
- Buttons are used to interact with the oscilloscope.



Labs 2: Results

- PL Hardware
- PS program
- Comblock Communication
- data transfer, that you plot

Optional:

Challenge

