JTAG/AVMM interface library

PAOLO DURANTE

Build Instructions

Build is tested on Enterprise Linux 9 only.

Firmware is tested on PCIe40 only for now (but JTAG library is device agnostic).

Low-level JTAG messaging is handled by libaji_client library from intel.

- git clone --recurse -b devel_aji https://gitlab.cern.ch/lhcb-daq40/lhcb-daq40-software
- cd pcie40_aji
- make

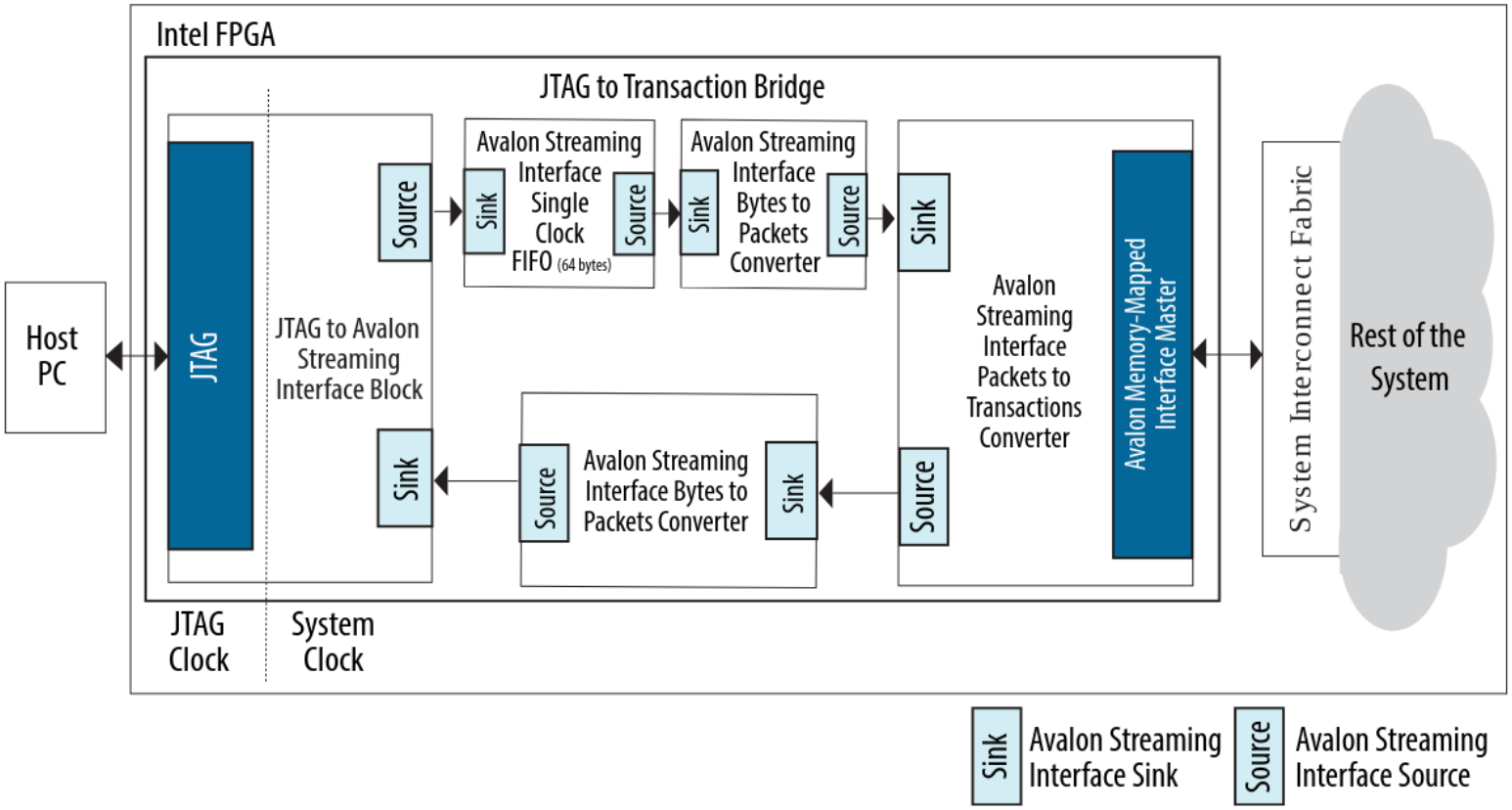
This will make a library (libpcie40_aji.a) and a command line application (pcie40_aji)

JTAG to AVMM bridge

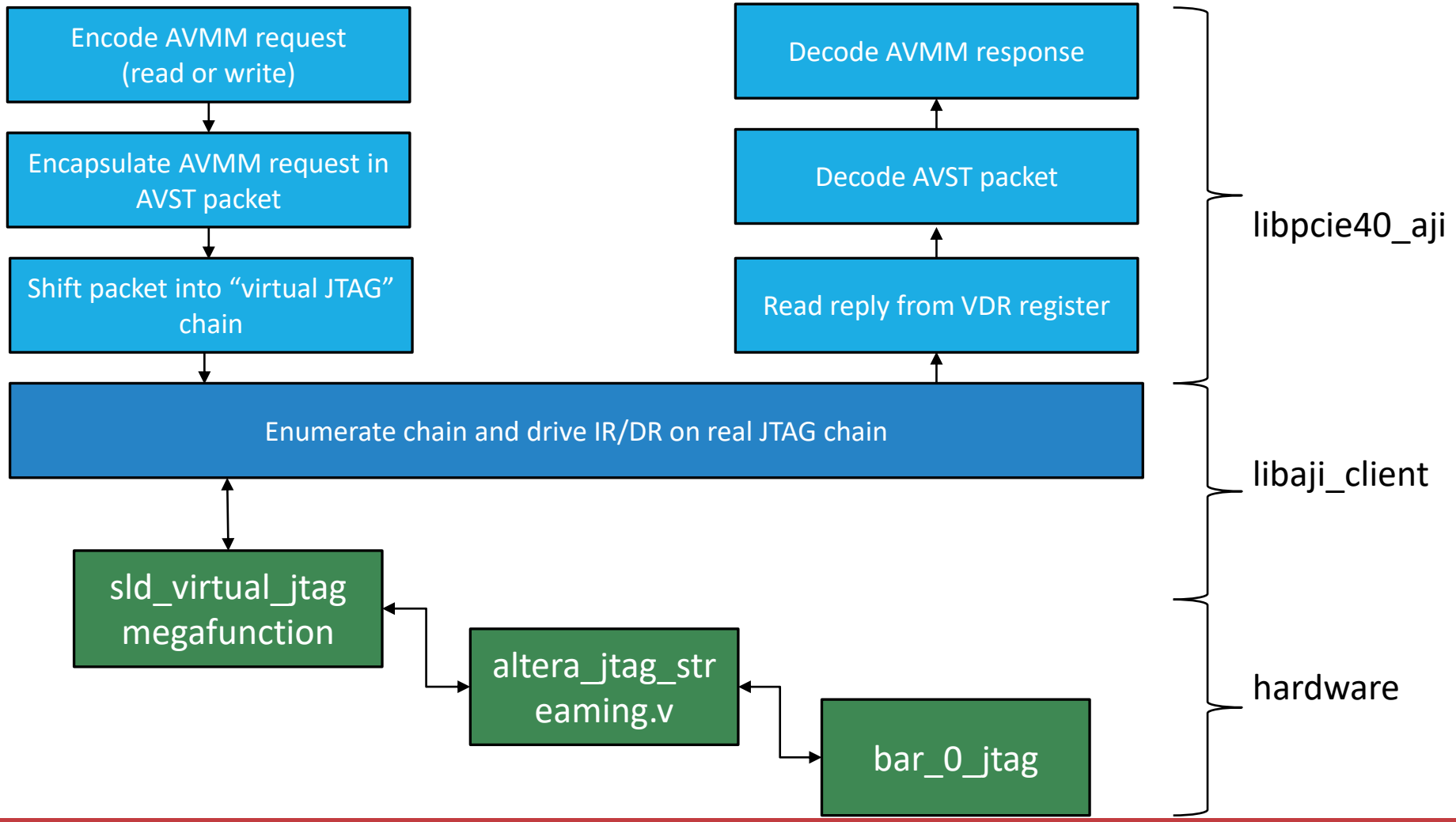
The protocol is mostly undocumented, except for a couple of images here:

https://www.intel.com/content/www/us/en/docs/programmable/683130/23-1/functional-description-49668.html

https://www.intel.com/content/www/us/en/docs/programmable/683130/23-1/functional-description-31334.html



Implementation



Programming Interface

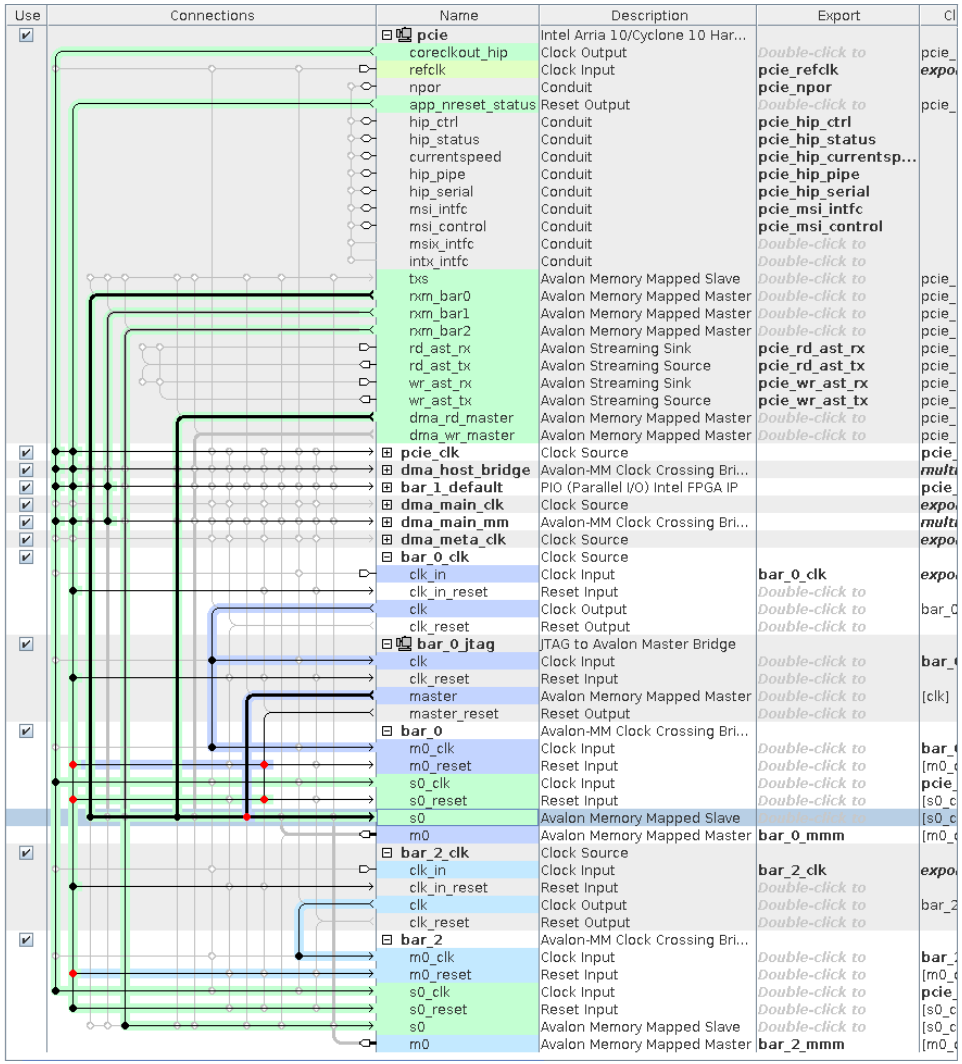
Main API is very simple: one function to read and one to write.

AJI_ERROR **aji_avmm_w32**(AJI_CHAIN_ID chain_id, AJI_OPEN_ID open_id, const AJI_HUB_INFO *hub, const AJI_HIER_ID *hier, int index, uint32_t addr, uint32_t data);

AJI_ERROR **aji_avmm_r32**(AJI_CHAIN_ID chain_id, AJI_OPEN_ID open_id, const AJI_HUB_INFO *hub, const AJI_HIER_ID *hier, int index, uint32_t addr, uint32_t *data);

However, in addition to the address and data, some JTAG chain information are required at well, these come from the Intel AJI client library and from the Quartus compilation reports.

Change to PCIe40 firmware



Use	Connections	Name	Description	Export	Cl
☑		⊟ pcie	Intel Arria 10/Cyclone 10 Har...		
		coreclkout_hip	Clock Output	Double-click to	pcie_
		refclk	Clock Input	pcie_refclk	expo
		npor	Conduit	pcie_npor	
		app_nreset_status	Reset Output	Double-click to	pcie_
		hip_ctrl	Conduit	pcie_hip_ctrl	
		hip_status	Conduit	pcie_hip_status	
		currentspeed	Conduit	pcie_hip_currentsp...	
		hip_pipe	Conduit	pcie_hip_pipe	
		hip_serial	Conduit	pcie_hip_serial	
		msi_intfc	Conduit	pcie_msi_intfc	
		msi_control	Conduit	pcie_msi_control	
		msix_intfc	Conduit	Double-click to	
		intx_intfc	Conduit	Double-click to	
		txs	Avalon Memory Mapped Slave	Double-click to	pcie_
		rxm_bar0	Avalon Memory Mapped Master	Double-click to	pcie_
		rxm_bar1	Avalon Memory Mapped Master	Double-click to	pcie_
		rxm_bar2	Avalon Memory Mapped Master	Double-click to	pcie_
		rd_ast_rx	Avalon Streaming Sink	pcie_rd_ast_rx	pcie_
		rd_ast_tx	Avalon Streaming Source	pcie_rd_ast_tx	pcie_
		wr_ast_rx	Avalon Streaming Sink	pcie_wr_ast_rx	pcie_
		wr_ast_tx	Avalon Streaming Source	pcie_wr_ast_tx	pcie_
		dma_rd_master	Avalon Memory Mapped Master	Double-click to	pcie_
		dma_wr_master	Avalon Memory Mapped Master	Double-click to	pcie_
☑		⊞ pcie_clk	Clock Source		pcie_
☑		⊞ dma_host_bridge	Avalon-MM Clock Crossing Bri...		*multi*
☑		⊞ bar_1_default	PIO (Parallel I/O) Intel FPGA IP		pcie_
☑		⊞ dma_main_clk	Clock Source		*expo*
☑		⊞ dma_main_mm	Avalon-MM Clock Crossing Bri...		*multi*
☑		⊞ dma_meta_clk	Clock Source		*expo*
☑		⊟ bar_0_clk	Clock Source		
		clk_in	Clock Input	bar_0_clk	*expo*
		clk_in_reset	Reset Input	Double-click to	
		clk	Clock Output	Double-click to	bar_0
		clk_reset	Reset Output	Double-click to	
☑		⊟ bar_0_jtag	JTAG to Avalon Master Bridge		
		clk	Clock Input	Double-click to	bar_
		clk_reset	Reset Input	Double-click to	
		master	Avalon Memory Mapped Master	Double-click to	[clk]
		master_reset	Reset Output	Double-click to	
☑		⊟ bar_0	Avalon-MM Clock Crossing Bri...		
		m0_clk	Clock Input	Double-click to	bar_
		m0_reset	Reset Input	Double-click to	[m0_
		s0_clk	Clock Input	Double-click to	pcie_
		s0_reset	Reset Input	Double-click to	[s0_c
		s0	Avalon Memory Mapped Slave	Double-click to	[s0_c
		m0	Avalon Memory Mapped Master	bar_0_mmm	[m0_
☑		⊟ bar_2_clk	Clock Source		
		clk_in	Clock Input	bar_2_clk	*expo*
		clk_in_reset	Reset Input	Double-click to	
		clk	Clock Output	Double-click to	bar_2
		clk_reset	Reset Output	Double-click to	
☑		⊟ bar_2	Avalon-MM Clock Crossing Bri...		
		m0_clk	Clock Input	Double-click to	bar_
		m0_reset	Reset Input	Double-click to	[m0_
		s0_clk	Clock Input	Double-click to	pcie_
		s0_reset	Reset Input	Double-click to	[s0_c
		s0	Avalon Memory Mapped Slave	Double-click to	[s0_c
		m0	Avalon Memory Mapped Master	bar_2_mmm	[m0_

New QSYS instance: **bar_0_qsys**

- JTAG to Avalon Master Bridge
- Connects to same slave interface as PCIe BAR0 master
- Same should be added for BAR2

We need to know the **sld instance index** to access it later (from Quartus):

Info (12133): Instantiated megafunction "pcie_top_x8x8:pcie_top|pcie_x8_wrapper:pcie_0|pcie_ax_x8_gen3_none_ecs:\gen_pcie:qsys_pcie|pcie_ax_x8_gen3_none_ecs_altera_jtag_avalon_master_181_cjcer3q:**bar_0_jtag**|altera_avalon_st_jtag_interface:jtag_phy_embedded_in_jtag_master|altera_jtag_sld_node:node|sld_virtual_jtag_basic:**sld_virtual_jtag_component**" with the following parameter:

Info (12134): Parameter "sld_mfg_id" = "110"
Info (12134): Parameter "sld_type_id" = "132"
Info (12134): Parameter "sld_version" = "1"
Info (12134): Parameter "sld_auto_instance_index" = "YES"
Info (12134): Parameter "sld_instance_index" = "0"
Info (12134): Parameter "sld_ir_width" = "3"
Info (12134): Parameter "sld_sim_action" = ""
Info (12134): Parameter "sld_sim_n_scan" = "0"
Info (12134): Parameter "sld_sim_total_length" = "0"

Example

LIBRARY

```
$ ./pcie40_aji -j 5-1.1.1
  (6) device_name=(null) hw_name=PCIe40 server=tdeb10 port=5-1.1.1 chain_id=0xc13d50
persistent_id=469762051, chain_type=1, features=34816, server_version_info_list=Version 22.1std.0 Build
915 10/25/2022 SC Standard Edition
    Number of devices on chain is 2
    (A1) device_id=02E660DD, instruction_length=10, features=4,
device_name=10AX115H1(.|E2|ES)/10AX115H2/..
            (B6) idcode 0x0C206E00 JTAG PHY #0

$ ./pcie40_aji -j 5-1.1.1 -i 6 -a 0x1000 -w 0x1234abcd # write value

$ ./pcie40_aji -j 5-1.1.1 -i 6 -a 0x1000 -r # read back
0x1234ABCD
```

SYSTEMCONSOLE

```
% master_read_32 {/devices/10AX115H1(.|E2|ES)|10AX115H2|..@1#5-
1.1.1#PCIe40#tdeb10/(link)/JTAG/alt_sld_fab_sldfabric.node_6/phy_0/bar_0_jtag.master} 0x1000 1
0x1234abcd

% master_write_32 {/devices/10AX115H1(.|E2|ES)|10AX115H2|..@1#5-
1.1.1#PCIe40#tdeb10/(link)/JTAG/alt_sld_fab_sldfabric.node_6/phy_0/bar_0_jtag.master} 0x1000 0x6789
```

LIBRARY

```
$ ./pcie40_aji -j 5-1.1.1 -i 6 -a 0x1000 -r
0x00006789
```