Assessing On-Chip High-Performance Interfaces on Xilinx and Intel FPGA-SoC Devices

Reconfigurable Computing 2019/2020

João Vieira

July 31, 2020

Abstract

Systems on Chip (SoCs) are devices that include several circuits with different functionalities cooperating to perform a given computational workload. For instance, Xilinx and Intel produce SoCs that integrate both hard *Central Processing Units* (CPUs) and *Field-Programmable Gate Arrays* (FPGAs), communicating through on-chip high-performance interfaces. Such interfaces usually allow for much higher data rates than off-chip connections, such as *Peripheral Component Interconnect* (PCI), and also lower latencies. The work proposed in this document aims at assessing the performance of the communication channels between the CPUs and the FPGA fabric of two SoC devices produced by Xilinx and Intel. The obtained performance measurements are compared with the theoretical bandwidth of the devices as advertised by the producing brands, and the methodology for reproducing the results is briefly explained.

1 Introduction

Field-Programmable Gate Array (FPGA)-based *Systems on Chip* (SoCs), also called FPGA-SoCs, are devices that combine hard *Central Processing Units* (CPUs) with reconfigurable logic. Such devices are useful for executing workloads that can benefit from hardware acceleration without resourcing to power-hungry *Graphic Processing Units* (GPUs) or fabricating custom *Application Specific Integrated Circuits* (ASICs).

FPGA-SoCs make it possible to efficiently offload certain phases of applications running on the hard CPUs to the reconfigurable logic. Both circuits are connected through on-chip high-performance communication channels capable of much higher bandwidths than common device-to-device interfaces, such as *Peripheral Component Interconnect* (PCI). For example, Xilinx and Intel both include in their low-end device families, ZYNQ-7000 and Cyclone V, respectively, three main types of interfaces between the *Processing System* (PS) (*Hard Processor System* (HPS) in Intel's notation) and the *Programmable Logic* (PL) (FPGA in Intel's notation): lightweight interfaces meant for low-throughput FPGA-implemented devices; high-performance interfaces for high-throughput accelerators; and *Accelerator Coherency Ports* (ACPs) for cache-coherent transactions. Table 1 summarizes the on-chip interfaces of Xilinx ZYNQ-7000 and Intel Cyclone V device families.

Table 1: On-chip interface comparison between the Xilinx ZYNQ-7000 and the Intel Cyclone V device families.

Xilinx ZYNQ-7000	Intel Cyclone V
***General-Purpose* (GP) ports:** There are two of these ports in the ZYNQ-7000 devices. They have a fixed width of 32 bit and no internal buffers, making them suitable for low-throughput applications.	***FPGA-to-HPS* (F2H):** This port has a configurable width of 32, 64 or 128 bit. Being suitable for lightweight communication, it resembles the GP ports of the ZYNQ-7000 devices.
***High-Performance* (HP) ports:** There are four of these ports in the ZYNQ-7000 devices. They have widths of either 32 or 64 bit and built-in *First-In-First-Outs* (FIFOs), making them suitable for high throughput applications.	***FPGA-to-SDRAM* (F2S):** Instead of offering four ports like the ZYNQ-7000's HP ports, Cyclone V has a single port which is directly connected to the memory controller. This port can be split into three independent AXI ports with a combined port width of up to 256 bit (e.g., 1×256 bit or $1 \times 128 + 2 \times 64$ bit).
***Accelerator Coherency Port* (ACP):** Additional 64-bit port that allows cache-coherent access to the memory. Performance-wise, this port resembles a HP port.	***Accelerator Coherency Port* (ACP):** This port matches the ACP port of the ZYNQ-7000 devices.

The work proposed in this document aims at assessing the performance of the on-chip high-performance interfaces present in Xilinx and Intel's low-end FPGA-SoC devices, namely the ZYNQ-7000 and the Cyclone V device families. For that purpose, efficient *Direct Memory Access* (DMA) engines connected to FPGA-implemented devices are used to stress the on-chip high-performance interfaces and allow measuring the maximum achieved data rates. All in all, the main results of this work are the following:

1. Architectures and *Register Transfer Level* (RTL) implementations of systems to evaluate the performance of the on-chip high-performance communication channels of Xilinx ZYNQ-7000 and Intel Cyclone V FPGA-SoC device families;
2. Evaluation and comparison of the on-chip communication channels of the SoC devices included in the Zybo board from Digilent (which features a Xilinx ZYNQ-7010 device) and the DE1-SoC board from TerasIC (featuring an Intel Cyclone V SE device).

The rest of this document is organized as follows: Section 2 presents the previous work; Section 3 briefly describes the proposed methodology and the architectures for stressing the on-chip high-performance interfaces and measuring the maximum allowed data rates. Section 4 explains important implementation details of the framework used for evaluating the Xilinx device and respective performance results. Section 5 analyzes the Intel device. Finally, Section 6 concludes this work.

2 Previous Work

Several efforts have been made towards analyzing the performance of FPGA-SoCs' memory hierarchies and on-chip communication channels. The first known results were presented by Sadri *et al.* [1]. They assessed the performance of the ACP in the Xilinx ZYNQ-7020 device, which, theoretically, is capable of the same data rate than a single HP port. Their results show that the ACP port achieves a duplex throughput of 1.7 GB/s when connected to a FPGA-implemented device operating at 125 MHz.

Sklyarov *et al.* [2] also focused their efforts in evaluating the high-performance interfaces between the PS and the PL on the Xilinx ZYNQ-7000 device family. Although they do not explicitly show the maximum bandwidth attained at 100 MHz, it can be derived from the results. Using a single 64-bit HP port at 100 MHz, their system achieved a maximum throughput of 284 MB/s, which is significantly lower than the theoretically possible 800 MB/s.

Also focusing the Xilinx ZYNQ-7000 device family, Tahghighi *et al.* [3] developed a mathematical model that allows estimating the latency of the memory accesses from the PL. Although their model considers several important parameters, it does not include the combination of several HP ports to increase the bandwidth, which limits its scope. Furthermore, it is limited to the Xilinx ZYNQ-7000 device family, not allowing to draw conclusions for different devices.

In addition to [1, 2, 3], there are other works aiming at assessing the performance of the on-chip communication channels of the Xilinx ZYNQ-7000 devices [4]. However, there has not been an equivalent effort to evaluate these interfaces on Intel FPGA-SoCs. Göbel *et al.* [5] presented one of the few studies that also evaluate the performance of the on-chip communication channels on Intel FPGA-SoCs. Their work focus on accessing the performance of the on-chip high-performance channels between the hard CPUs and the reconfigurable logic on both Xilinx and Intel low-end device families (ZYNQ-7000 and Cyclone V, respectively). Their assessment is divided into two phases. First, they stress the on-chip high-performance communication channels of the devices and register the maximum achieved data rates. Second, they assess the channels' performance in the context of video processing applications. The expected PS-to-PL (HPS-to-FPGA in Intel's notation) bandwidth of the high-performance channels and the results obtained in their experiments are summarized in Table 2.

In their experiments, Göbel *et al.* were able to almost achieve the theoretical maximum bandwidth provided

Table 2: Results obtained by Göbel *et al.* [5] regarding the bandwidth of the high-performance on-chip communication channels of two Xilinx and Intel's FPGA-SoCs (ZYNQ-7020 and Cyclone V SE, respectively).

		Xilinx ZYNQ-7000	**Intel Cyclone V**
DRAM	**Bandwidth** [MT/s]	**1066**	**800**
	Transfer width [bit]	32	32
	Bandwidth [MB/s]	**4264**	**3200**
AXI3 Interface Frequency [MHz]		110	110
Theoretical Bandwidth [MB/s]	**1 × HP @** 110 MHz	880	880
	2 × HP @ 110 MHz	1760	1760
	4 × HP @ 110 MHz	**3520**	**3200**
Real Bandwidth [MB/s]	**1 × HP @** 110 MHz	879	879
	2 × HP @ 110 MHz	1723	1723
	4 × HP @ 110 MHz	**3499**	**2715**

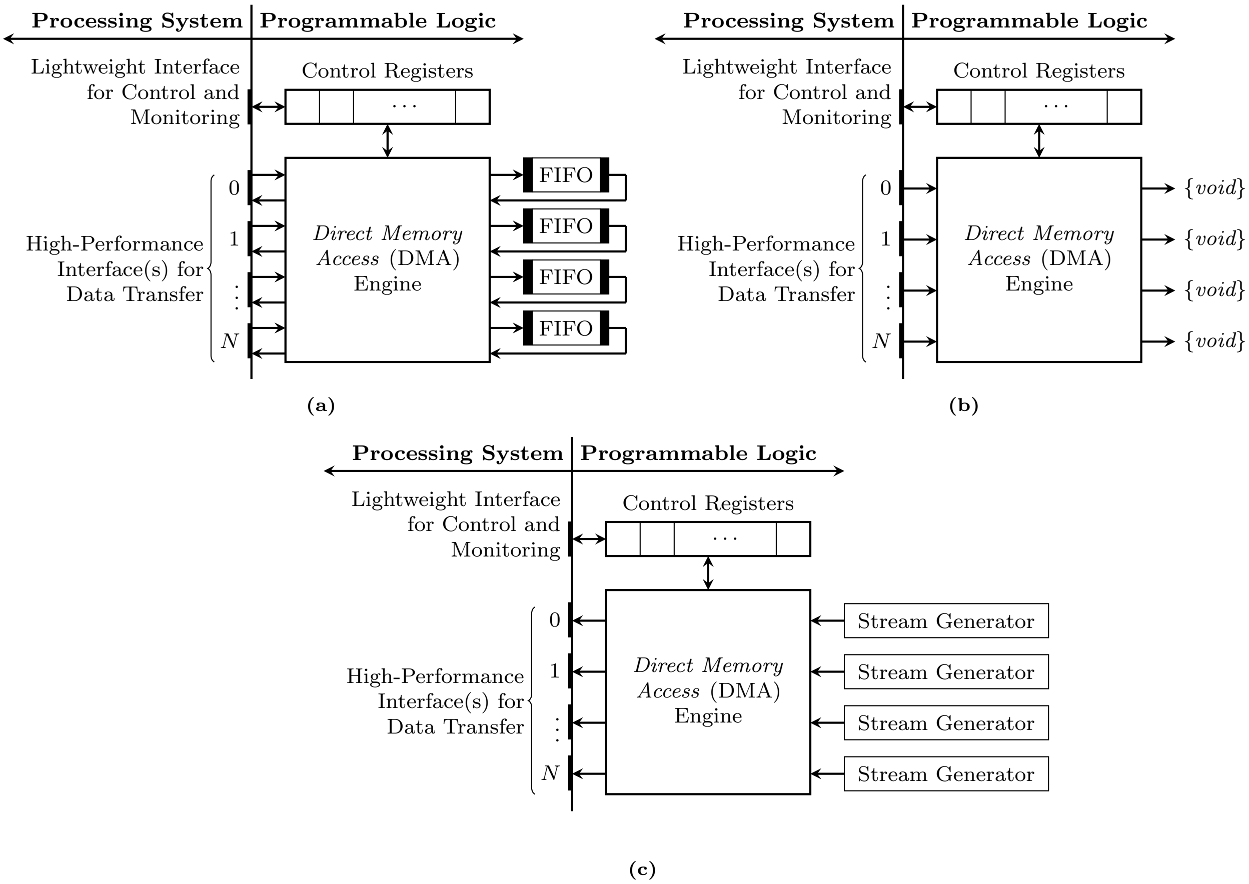

Processing System
Programmable Logic
Lightweight Interface for Control and Monitoring
Control Registers
...
High-Performance Interface(s) for Data Transfer
0
1
N
Direct Memory Access (DMA) Engine
FIFO
FIFO
FIFO
FIFO
(a)
Processing System
Programmable Logic
Lightweight Interface for Control and Monitoring
Control Registers
...
High-Performance Interface(s) for Data Transfer
0
1
N
Direct Memory Access (DMA) Engine
{void}
{void}
{void}
{void}
(b)
Processing System
Programmable Logic
Lightweight Interface for Control and Monitoring
Control Registers
...
High-Performance Interface(s) for Data Transfer
0
1
N
Direct Memory Access (DMA) Engine
Stream Generator
Stream Generator
Stream Generator
Stream Generator
(c)

Figure 1: Systems proposed to measure the (a) duplex, (b) PS-to-PL, and (c) PL-to-PS bandwidths of the FPGA-SoC devices' on-chip high-performance interfaces.

by the four HP ports of the Xilinx device for the testing frequency (110 MHz). However, they could not achieve the theoretical maximum data rate of the 256-bit F2S port in the Intel Cyclone V device. They explain that this effect is probably related to the block size used for the assessment. It is also noticeable that while the maximum throughput of the HP ports in the ZYNQ-7000 device is bounded by the AXI3 interface operation frequency, the F2S interface of the Cyclone V device is bounded by the SDRAM maximum bandwidth. This is caused by the lower data rate of the SDRAM connected to the Cyclone V device (800 MT/s) compared to the one connected to the ZYNQ-7000 device (1066 MT/s).

As the work of Göbel *et al.* is the most comprehensive study on the performance of the on-chip high-performance interfaces that considers both Xilinx and Intel FPGA-SoCs at the date this document is written, it will be used as a reference.

To simplify the notation used in the rest of this document, whenever both Xilinx and Intel devices are referred, only the Xilinx terms will be explicitly written. For example, "PS-to-PL (HPS-to-FPGA in Intel's notation) bandwidth" will be simply referred to as "PS-to-PL bandwidth".

The next Section briefly explains the evaluation methodology used in this work.

3 Methodology

To evaluate the duplex, PS-to-PL, and PL-to-PS bandwidths of the on-chip high-performance communication channels of the Xilinx and Intel's FPGA-SoCs, three simple systems were envisioned. These architectures consist of one or more DMA engines connected to the high-performance ports (HP on the Xilinx device and F2S on the Intel device), their respective control interfaces connected to a lightweight port (GP on Xilinx devices and F2H on Intel devices), and some extra hardware to either loop data back, absorb an incoming stream or produce a data stream on the reconfigurable logic. The top-level architectures of the proposed systems are depicted in Figure 1.

For comparison purposes, this work uses devices of the same family than those used in [5]. More specifically, the Xilinx ZYNQ-7010 (Zybo board from Digilent [6]) and the Intel Cyclone V SE (DE1-SoC from TerasIC [7]). The testing systems were developed using *VHSIC Hardware Description Language* (VHDL) language and were targeted in both the Xilinx and the Intel devices with the same operating frequency to ensure a fair comparison.

In the software-side, simple programs will be written in C language and targeted in both devices to control the hardware structures in the reconfigurable logic and output the results of the experiment.

In summary, the main objectives of this project are:

1. Learning to use Xilinx and Intel's platforms for FPGA and FPGA-SoC development (Xilinx Vivado and Intel Quartus Prime, respectively);
2. Researching architectures of DMA devices capable of stressing the on-chip high-performance communication interfaces of the FPGA-SoC devices;
3. Producing a system capable of evaluating the performance of the on-chip high-performance communication interfaces of the FPGA-SoC devices, as well as supporting software;
4. Comparing the obtained results with the theoretical limits of the devices and the results of previous studies.

4 Xilinx ZYNQ-7010 device

The three systems proposed in Section 3 were implemented and targeted in the ZYNQ-7010 device to assess the performance of the HP ports. Due to the existence of four distinct HP ports, multiple DMA engines were implemented in the PL. The used DMA engine was the regular one that comes with Xilinx Vivado 2018.3 since it is capable of fully exploiting the bandwidth of the on-chip high-performance interfaces [8]. Additionally, two simple circuits were implemented to absorb the streams exported by the DMA engines and to produce streams to be consumed. These two circuits were used in the systems to evaluate the PS-to-PL and PL-to-PS bandwidths, respectively. While the circuit implemented to absorb the streams produced by the DMA engines consists of a single constant connected the *ready* signal of the AXI4-Stream master interface [9], the circuit to generate data streams is more complex, consisting of a *Finite-State Machine* (FSM) that generates a predetermined number of words, one per cycle. The system that evaluates the duplex bandwidth features twice as many DMA engines as the other two systems. In that system, the DMA engines are paired and connected through a FIFO, as shown in Figure 2a. Figures 2b and 2c illustrate the architectures of the systems to evaluate the PS-to-PL and PL-to-PS bandwidths, respectively.

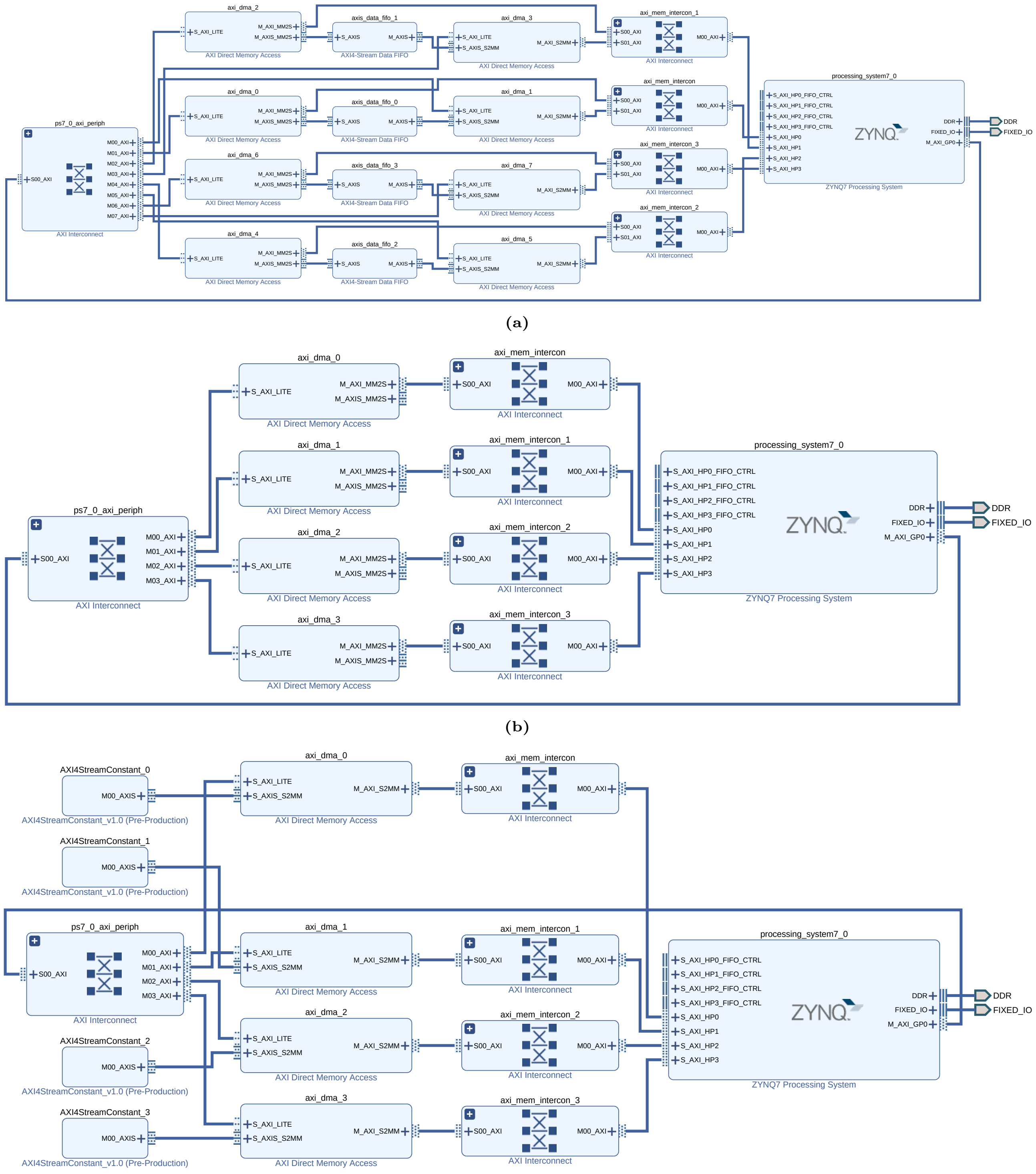
To control and monitor the data transfers through the DMA engines and calculate their performance, simple bare-metal applications were written using Xilinx hardware libraries and targeted in a single core of the dual-core ARM Cortex-A9 included in the ZYNQ-7010 device.

4.1 Experimental Results

The three designs were synthesized and implemented using Xilinx Vivado 2018.3 with a PL operation frequency of 100 MHz. Additionally, the system depicted in Figure 2b was also synthesized for an operation frequency of 150 MHz, in an attempt to reproduce the results obtained by Göbel *et al.* regarding the maximum PS-to-PL bandwidth. All the Xilinx DMA devices were configured for a maximum burst size of sixteen, and a maximum stream size of 64 MiB. The four ZYNQ HP ports were tested for both 32 and 64-bit configurations. The hardware requirements of the implemented systems are listed in Table 6, in Appendix A.

Table 3 summarizes the performance results obtained using the three implemented systems. The rows painted in yellow represent configurations in which the bandwidth is degraded compared to the theoretical values. To understand why these configurations lead to sub-optimal bandwidth utilization, the low-level architecture of the ZYNQ-7000 device family has to be considered. As documented in [10], page 63, the HP ports 0 and 1, as well as ports 2 and 3, share the same interconnect to the DDR controller. Naturally, using both HP ports 0 and 1 in a 64-bit duplex configuration requires to multiplex a single channel to the DDR controller, leading to a sub-optimal memory bandwidth utilization. However, when using HP ports 0 and 2 in the same configuration, two distinct channels to the DDR controller are used, leading to an optimal memory bandwidth utilization close to the maximum theoretical value. It is also worth mentioning the abnormally high standard deviation associated with the sub-optimal configurations, which is most likely due to the entropy generated at the memory controller level. A premise that supports this conclusion is the fact that using HP ports 0, 1, and 2 in a 64-bit duplex configuration leads to a lower data rate than using only HP ports 0 and 2 in the same configuration.

Although the previous experiments allowed to fully exploit the bandwidth of the on-chip high-performance interfaces in duplex mode, they were insufficient to reproduce the results obtained by Göbel *et al.* regarding the PS-to-PL bandwidth of the Xilinx device. In order to increase even further the bandwidth requirement on the PL side, the system to evaluate the PS-to-PL bandwidth was re-implemented for an operating frequency of 150 MHz. With the new operation frequency, the PL component of the system is capable of a maximum data rate of 4800 MB/s, which is even higher than the DDR maximum bandwidth. The results of this experiment are shown in Table 4. As expected, the maximum achieved bandwidth surpasses that of the same system implemented for an operating frequency of 100 MHz. The configuration that leads, in average, to the highest data rate (3448.33 MB/s) is the one using HP ports 0, 1 and 2. The maximum achieved data rate is compatible with the conclusions of Göbel *et al.* Furthermore, in one or more experiments using all the four HP ports, a bandwidth



(a)

(b)

(c)

Figure 2: Block designs of the systems to evaluate the (a) duplex, (b) PS-to-PL, and (c) PL-to-PS bandwidths of the on-chip high-performance interfaces of the Xilinx device.

higher than 4 GB/s was obtained. However, such performance is not always possible due to limitations at the memory controller level. It is also worth mentioning that, under stress, the HP ports achieve higher data rates on single-sided transfers than on duplex transfers.

To conclude the assessment of the ZYNQ-7000 device, another scenario based on the system to evaluate the duplex bandwidth was considered in which streams of multiple sizes were transferred from the PS to the PL and back to the PS. For this experiment, a single 64-bit HP port was used, the PL operation frequency was set to 100 MHz, and each stream size was tested 200 times. Figure 3 illustrates the results of this experiment. Results show that small stream sizes only achieve a fraction of the theoretical maximum bandwidth allowed by the used configuration. Only for streams bigger than 64 KiB a real bandwidth of more than 90% of the theoretical maximum is achieved.

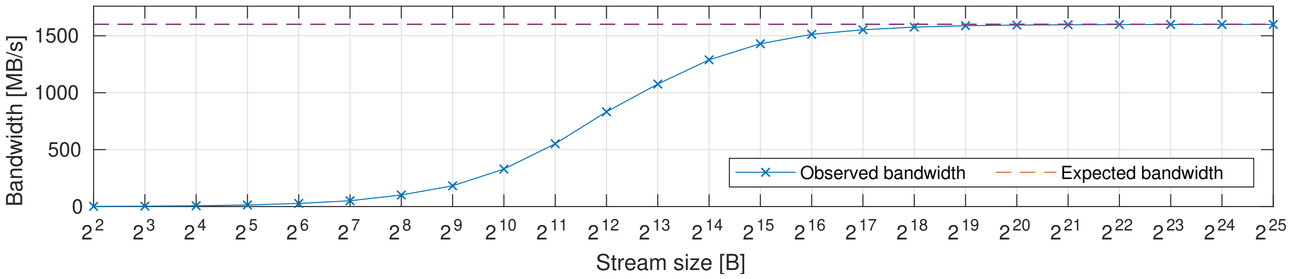


Figure 3: Observed duplex bandwidth for different stream sizes. The system was designed using a single 64-bit HP port and a loop-back FIFO, allowing to fully exploit the duplex bandwidth of the HP port. The PL operation frequency is 100 MHz. Each data point represents the average of 200 transactions of the same size. The maximum standard deviation was 5.70 MB/s.

Table 3: Expected and observed bandwidth and respective system efficiency for several system configurations using HP ports of the ZYNQ-7010 device. Each configuration was tested 200 times for 32 MiB data blocks. The PL operating frequency was 100 MHz.

		Channels	**Bandwidth** [MB/s]					**Efficiency** [%]
			Expected	**Observed**				
				Average	**Minimum**	**Maximum**	σ	
32 bit per channel	**Duplex**	**HP0**	800.00	799.95	799.95	799.96	0.0003	99.99
		HP0,1	1600.00	1599.84	1599.84	1599.85	0.0017	99.99
		HP0,2	1600.00	1599.83	1599.79	1599.84	0.0085	99.99
		HP0,1,2	2400.00	2288.49	2280.15	2293.17	1.2566	95.35
		HP0,1,2,3	3200.00	2773.13	2697.44	2789.70	15.0109	86.66
	PS to PL	**HP0**	400.00	399.99	399.99	399.99	0.0001	100.00
		HP0,1	800.00	799.96	799.96	799.96	0.0006	100.00
		HP0,2	800.00	799.96	799.96	799.96	0.0006	100.00
		HP0,1,2	1200.00	1199.91	1199.91	1199.92	0.0010	99.99
		HP0,1,2,3	1600.00	1599.85	1599.85	1599.8521	0.0002	99.99
	PL to PS	**HP0**	400.00	399.99	399.99	400.0000	0.0078	100.00
		HP0,1	800.00	799.98	799.96	800.0000	0.0229	100.00
		HP0,2	800.00	799.98	799.96	800.0000	0.0228	100.00
		HP0,1,2	1200.00	1199.95	1199.91	1200.0000	0.0493	100.00
		HP0,1,2,3	1600.00	1599.92	1599.85	1600.0000	0.0800	100.00
64 bit per channel	**Duplex**	**HP0**	1600.00	1599.82	1599.81	1599.82	0.0054	99.99
		HP0,1	3200.00	2236.12	2212.38	2267.40	6.3216	69.88
		HP0,2	3200.00	3197.63	3193.18	3198.68	0.8462	99.93
		HP0,1,2	4264.00	2659.35	2639.86	2682.41	4.1705	62.37
		HP0,1,2,3	4264.00	2675.65	2108.10	2969.61	297.1049	62.75
	PS to PL	**HP0**	800.00	799.96	799.95	799.96	0.0012	99.99
		HP0,1	1600.00	1599.84	1599.83	1599.84	0.0018	99.99
		HP0,2	1600.00	1599.84	1599.83	1599.85	0.0016	99.99
		HP0,1,2	2400.00	2399.66	2399.64	2399.66	0.0054	99.99
		HP0,1,2,3	3200.00	3184.87	3103.57	3199.38	16.7246	99.53
	PL to PS	**HP0**	800.00	799.98	799.95	800.00	0.0286	100.00
		HP0,1	1600.00	1599.91	1599.83	1600.00	0.0917	99.99
		HP0,2	1600.00	1599.91	1599.83	1600.00	0.0914	100.00
		HP0,1,2	2400.00	2399.83	2399.63	2400.00	0.1895	99.99
		HP0,1,2,3	3200.00	3199.70	3199.39	3200.00	0.3170	99.99

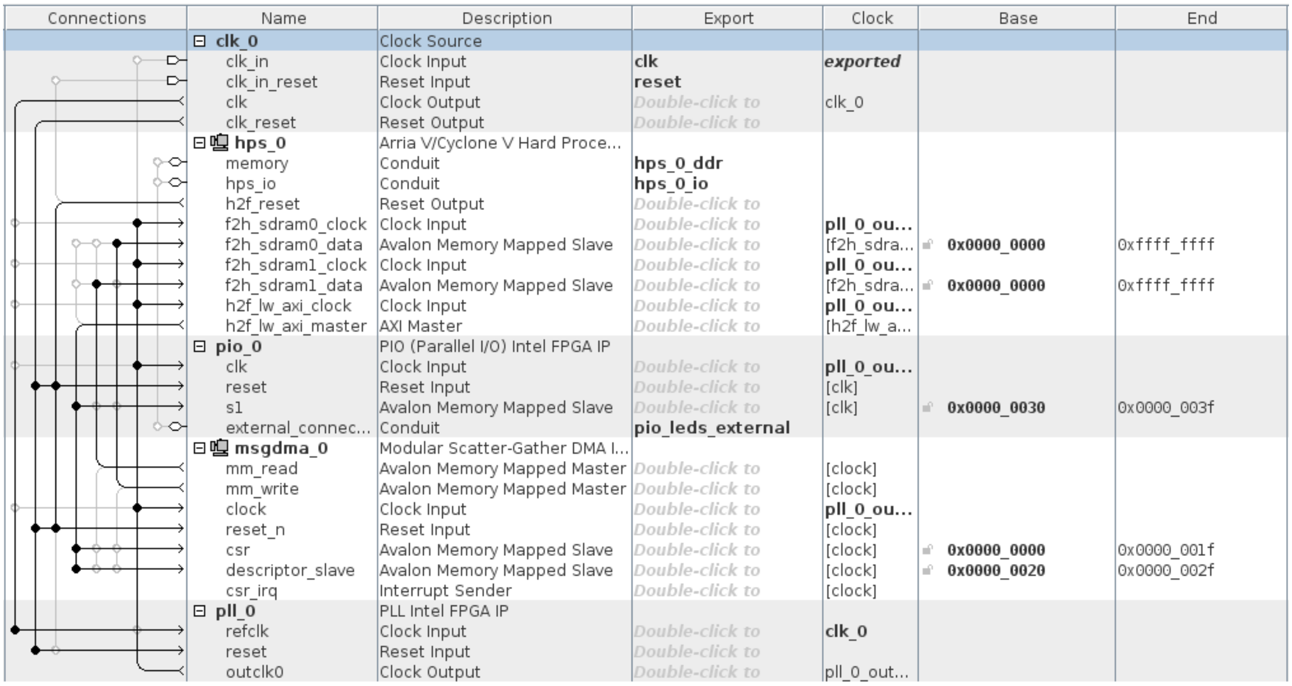
Table 4: Expected and observed bandwidth and respective system efficiency for configurations aiming at exploiting the maximum PS-to-PL bandwidth. Each configuration was tested 200 times for 32 MiB data blocks. The PL operating frequency was 150 MHz.

	Channels	**Bandwidth** [MB/s]					**Efficiency** [%]
		Expected	**Observed**				
			Average	**Minimum**	**Maximum**	σ	
64 bit/ channel, PS to PL	**HP0**	1200	1199.91	1199.91	1199.91	0.0005	99.99
	HP0,1	2400	2393.39	2371.02	2397.02	2.3698	99.72
	HP0,2	2400	2396.52	2396.07	2398.13	0.2730	99.86
	HP0,1,2	3600	3448.33	3413.08	3510.90	11.2324	95.79
	HP0,1,2,3	4264	3154.97	2939.33	4001.69	133.1674	73.99

5 Intel Cyclone V SE device

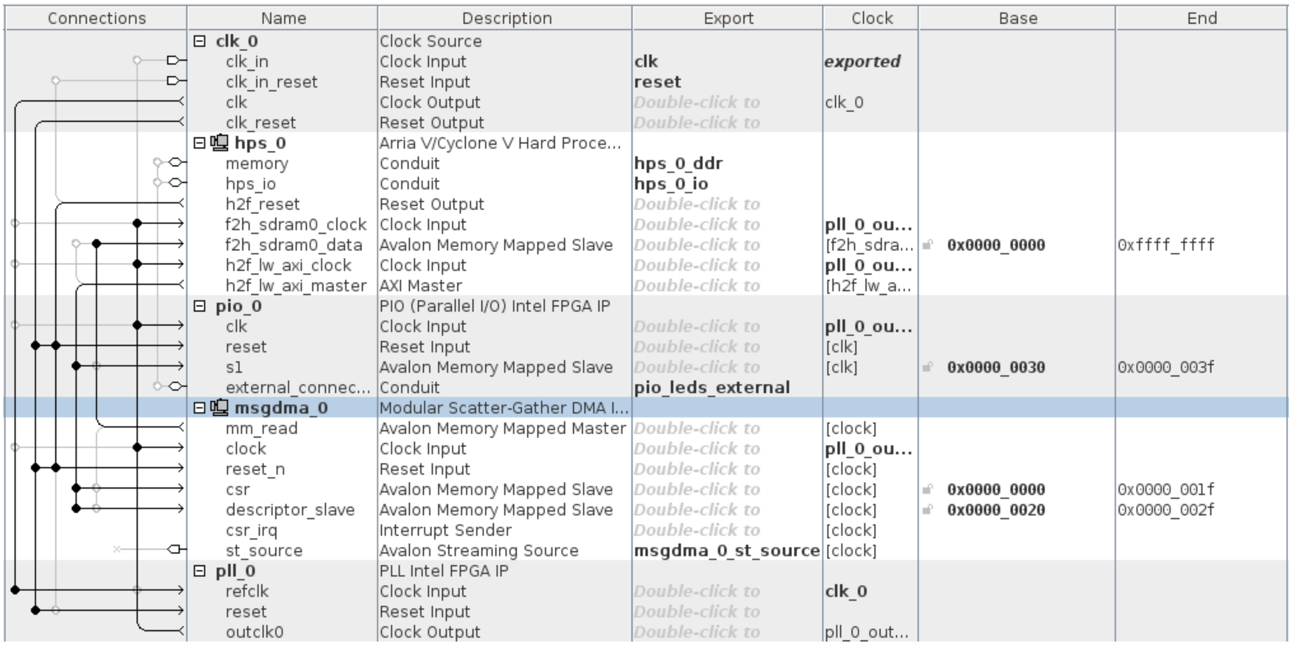
Similarly to the assessment of the ZYNQ-7010 device, the same three systems depicted in Figure 1 were implemented and targeted in the Cyclone V SE device. Since the Cyclone V device family only has one on-chip high-performance port that connects the FPGA to the DDR controller through the HPS, a single DMA engine was used in each system. The used DMA engine was Intel's *Modular Scatter-Gather Direct Memory Access* (MSGDMA), which revealed capable of fully-exploiting the bandwidth of the F2S interface. The Intel's MSGDMA allows three configurations regarding its input and output interfaces: AXI3-to-AXI3, AXI3-to-Avalon-Stream, and Avalon-Stream-to-AXI3. Furthermore, the MSGDMA also includes an internal data FIFO, which allows simplifying the system to assess the duplex bandwidth by using the MSGDMA in the configuration AXI3-to-AXI3 and connecting both interfaces to the F2S port directly, as shown in Figure 4. Note that both the AXI3 interfaces of the MSGDMA can be connected to the F2S port even if they have 256 bit each. This is only possible because each of these interfaces is single-sided (one interface only reads and the other only writes), and the F2S port is full-duplex. For testing the HPS-to-FPGA and the FPGA-to-HPS bandwidths, the MSGDMA was configured as AXI3-to-Avalon-Stream and Avalon-Stream-to-AXI3, respectively. Additionally, two simple circuits were developed to absorb the stream generated by the MSGDMA in the configuration AXI3-to-Avalon-Stream and generate a data stream to be absorbed by the MSGDMA in the configuration Avalon-Stream-to-AXI3. In resemblance to the system targeting the Xilinx device, the circuit to absorb the stream consists of a single wire connected to the *ready* signal of the Avalon-Stream master interface. On the other hand, the circuit to produce a data stream to be transmitted to the MSGDMA engine is simpler than its Xilinx counterpart. The streaming protocol used by the Xilinx DMA device is the AXI4-Stream protocol, which features signal indicating the end of the stream. Therefore, a FSM is needed to count the elements of the stream and activate the *last* signal during the transmission of the stream's last element. However, the streaming protocol used by Intel's MSGDMA (Avalon-Stream [11]) does not have a signal indicating the end of the stream. Thus, there is no need for a FSM. To produce a constant stream, it only takes to set the *data* bus to a constant and the *valid* signal to high. Figures 5 and 6 illustrate the systems to evaluate the HPS-to-FPGA and FPGA-to-HPS bandwidths, respectively.

In contrast with the software framework for controlling and monitoring the FPGA-implemented circuits in the Xilinx device (which was bare-metal), the software used to control the FPGA components on the Intel device was developed to be executed in Linux environment. This implementation choice had to do with the poorer bare-metal development support provided by Intel tools when comparing to Xilinx's more powerful and automated tools. For instance, compiling a bare-metal application for Intel devices requires the developer to write a linker script by hand (there is no tool to generate the linker script automatically). Although Intel provides linker scripts that can be used out-of-the-box, they only work for certain applications. Another possibility is to use the ARM compiler, instead of the default Intel bare-metal compiler, which requires the user to write a scatter file. Although scatter files are usually much less verbose than linker scripts, their syntax is also complex.



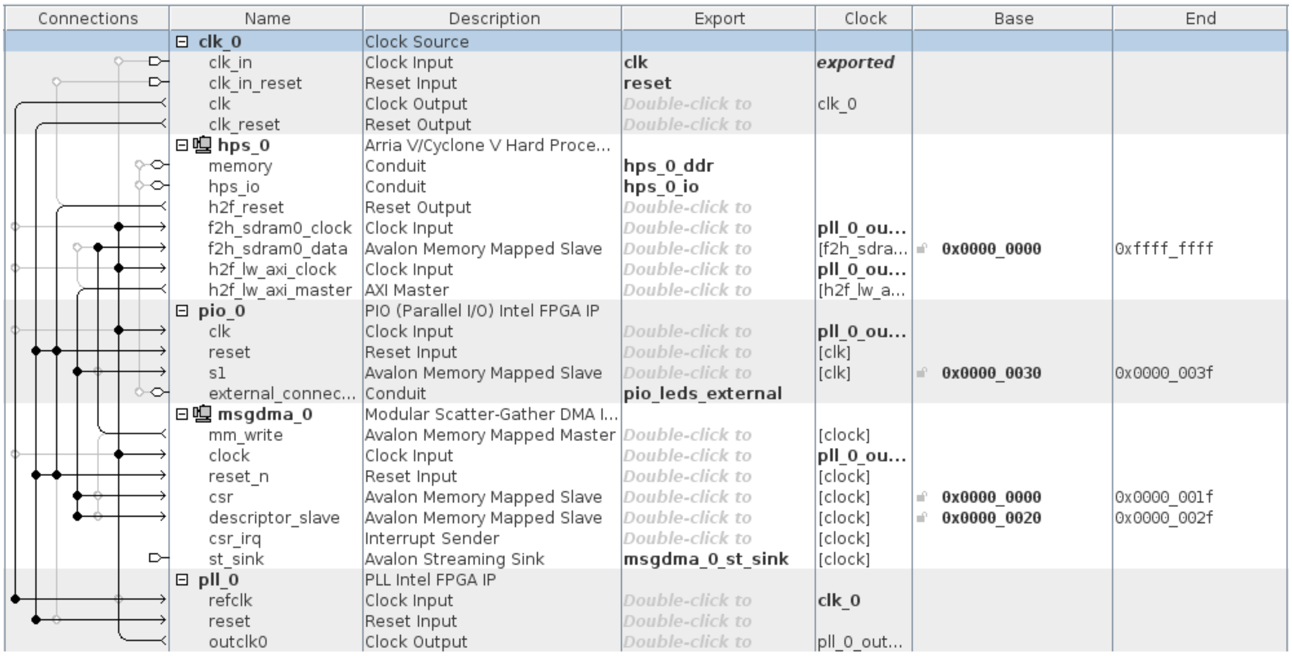
Connections	Name	Description	Export	Clock	Base	End
	⊟ **clk_0**	Clock Source				
	clk_in	Clock Input	**clk**	***exported***		
	clk_in_reset	Reset Input	**reset**			
	clk	Clock Output	*Double-click to*	clk_0		
	clk_reset	Reset Output	*Double-click to*			
	⊟ **hps_0**	Arria V/Cyclone V Hard Proce...				
	memory	Conduit	**hps_0_ddr**			
	hps_io	Conduit	**hps_0_io**			
	h2f_reset	Reset Output	*Double-click to*			
	f2h_sdram0_clock	Clock Input	*Double-click to*	**pll_0_ou...**		
	f2h_sdram0_data	Avalon Memory Mapped Slave	*Double-click to*	[f2h_sdra...	**0x0000_0000**	0xffff_ffff
	f2h_sdram1_clock	Clock Input	*Double-click to*	**pll_0_ou...**		
	f2h_sdram1_data	Avalon Memory Mapped Slave	*Double-click to*	[f2h_sdra...	**0x0000_0000**	0xffff_ffff
	h2f_lw_axi_clock	Clock Input	*Double-click to*	**pll_0_ou...**		
	h2f_lw_axi_master	AXI Master	*Double-click to*	[h2f_lw_a...		
	⊟ **pio_0**	PIO (Parallel I/O) Intel FPGA IP				
	clk	Clock Input	*Double-click to*	**pll_0_ou...**		
	reset	Reset Input	*Double-click to*	[clk]		
	s1	Avalon Memory Mapped Slave	*Double-click to*	[clk]	**0x0000_0030**	0x0000_003f
	external_connec...	Conduit	**pio_leds_external**			
	⊟ **msgdma_0**	Modular Scatter-Gather DMA I...				
	mm_read	Avalon Memory Mapped Master	*Double-click to*	[clock]		
	mm_write	Avalon Memory Mapped Master	*Double-click to*	[clock]		
	clock	Clock Input	*Double-click to*	**pll_0_ou...**		
	reset_n	Reset Input	*Double-click to*	[clock]		
	csr	Avalon Memory Mapped Slave	*Double-click to*	[clock]	**0x0000_0000**	0x0000_001f
	descriptor_slave	Avalon Memory Mapped Slave	*Double-click to*	[clock]	**0x0000_0020**	0x0000_002f
	csr_irq	Interrupt Sender	*Double-click to*	[clock]		
	⊟ **pll_0**	PLL Intel FPGA IP				
	refclk	Clock Input	*Double-click to*	**clk_0**		
	reset	Reset Input	*Double-click to*			
	outclk0	Clock Output	*Double-click to*	pll_0_out...		

Figure 4: Schematic of the system to assess the duplex bandwidth of the on-chip high-performance interfaces of the Intel device, as shown in the Platform Designer tool of Intel Quartus Prime.



Connections	Name	Description	Export	Clock	Base	End
	⊟ **clk_0**	Clock Source				
	clk_in	Clock Input	**clk**	***exported***		
	clk_in_reset	Reset Input	**reset**			
	clk	Clock Output	*Double-click to*	clk_0		
	clk_reset	Reset Output	*Double-click to*			
	⊟ **hps_0**	Arria V/Cyclone V Hard Proce...				
	memory	Conduit	**hps_0_ddr**			
	hps_io	Conduit	**hps_0_io**			
	h2f_reset	Reset Output	*Double-click to*			
	f2h_sdram0_clock	Clock Input	*Double-click to*	**pll_0_ou...**		
	f2h_sdram0_data	Avalon Memory Mapped Slave	*Double-click to*	[f2h_sdra...	**0x0000_0000**	0xffff_ffff
	h2f_lw_axi_clock	Clock Input	*Double-click to*	**pll_0_ou...**		
	h2f_lw_axi_master	AXI Master	*Double-click to*	[h2f_lw_a...		
	⊟ **pio_0**	PIO (Parallel I/O) Intel FPGA IP				
	clk	Clock Input	*Double-click to*	**pll_0_ou...**		
	reset	Reset Input	*Double-click to*	[clk]		
	s1	Avalon Memory Mapped Slave	*Double-click to*	[clk]	**0x0000_0030**	0x0000_003f
	external_connec...	Conduit	**pio_leds_external**			
	⊟ **msgdma_0**	Modular Scatter-Gather DMA I...				
	mm_read	Avalon Memory Mapped Master	*Double-click to*	[clock]		
	clock	Clock Input	*Double-click to*	**pll_0_ou...**		
	reset_n	Reset Input	*Double-click to*	[clock]		
	csr	Avalon Memory Mapped Slave	*Double-click to*	[clock]	**0x0000_0000**	0x0000_001f
	descriptor_slave	Avalon Memory Mapped Slave	*Double-click to*	[clock]	**0x0000_0020**	0x0000_002f
	csr_irq	Interrupt Sender	*Double-click to*	[clock]		
	st_source	Avalon Streaming Source	**msgdma_0_st_source**	[clock]		
	⊟ **pll_0**	PLL Intel FPGA IP				
	refclk	Clock Input	*Double-click to*	**clk_0**		
	reset	Reset Input	*Double-click to*			
	outclk0	Clock Output	*Double-click to*	pll_0_out...		

Figure 5: Schematic of the system to assess the HPS-to-FPGA bandwidth of the on-chip high-performance interfaces of the Intel device, as shown in the Platform Designer tool of Intel Quartus Prime.



Connections	Name	Description	Export	Clock	Base	End
	⊟ **clk_0**	Clock Source				
	clk_in	Clock Input	**clk**	***exported***		
	clk_in_reset	Reset Input	**reset**			
	clk	Clock Output	*Double-click to*	clk_0		
	clk_reset	Reset Output	*Double-click to*			
	⊟ **hps_0**	Arria V/Cyclone V Hard Proce...				
	memory	Conduit	**hps_0_ddr**			
	hps_io	Conduit	**hps_0_io**			
	h2f_reset	Reset Output	*Double-click to*			
	f2h_sdram0_clock	Clock Input	*Double-click to*	**pll_0_ou...**		
	f2h_sdram0_data	Avalon Memory Mapped Slave	*Double-click to*	[f2h_sdra...	**0x0000_0000**	0xffff_ffff
	h2f_lw_axi_clock	Clock Input	*Double-click to*	**pll_0_ou...**		
	h2f_lw_axi_master	AXI Master	*Double-click to*	[h2f_lw_a...		
	⊟ **pio_0**	PIO (Parallel I/O) Intel FPGA IP				
	clk	Clock Input	*Double-click to*	**pll_0_ou...**		
	reset	Reset Input	*Double-click to*	[clk]		
	s1	Avalon Memory Mapped Slave	*Double-click to*	[clk]	**0x0000_0030**	0x0000_003f
	external_connec...	Conduit	**pio_leds_external**			
	⊟ **msgdma_0**	Modular Scatter-Gather DMA I...				
	mm_write	Avalon Memory Mapped Master	*Double-click to*	[clock]		
	clock	Clock Input	*Double-click to*	**pll_0_ou...**		
	reset_n	Reset Input	*Double-click to*	[clock]		
	csr	Avalon Memory Mapped Slave	*Double-click to*	[clock]	**0x0000_0000**	0x0000_001f
	descriptor_slave	Avalon Memory Mapped Slave	*Double-click to*	[clock]	**0x0000_0020**	0x0000_002f
	csr_irq	Interrupt Sender	*Double-click to*	[clock]		
	st_sink	Avalon Streaming Sink	**msgdma_0_st_sink**	[clock]		
	⊟ **pll_0**	PLL Intel FPGA IP				
	refclk	Clock Input	*Double-click to*	**clk_0**		
	reset	Reset Input	*Double-click to*			
	outclk0	Clock Output	*Double-click to*	pll_0_out...		

Figure 6: Schematic of the system to assess the FPGA-to-HPS bandwidth of the on-chip high-performance interfaces of the Intel device, as shown in the Platform Designer tool of Intel Quartus Prime.

Naturally, the examples of scatter files provided by ARM also have a limited scope. In contrast with bare-metal, Linux development is much more powerful and there are simple known techniques to overcome issues such as addressing memory-mapped peripherals through their physical addresses, or reserve part of the DDR memory to be used directly by DMA engines without needing to translate virtual addresses. As explained by Kashani *et al.* [12], the physical addresses of the memory-mapped peripherals can be accessed from the Linux system using the function `mmap` to map those addresses into addresses in the application addressing space. To reserve part of the DDR to be used directly by DMA engines, it is possible to assign only a sub-region of the DDR to the Linux kernel at boot time, leaving the upper addresses free. By default, the upper part of the DDR that is not used by the Linux kernel is ignored by the operating system and is initialized as non-cacheable memory. However, it can still be accessed using the same method to address the memory-mapped peripherals, using the `mmap` function to map the physical addresses of the upper DDR region into addresses in the application addressing space.

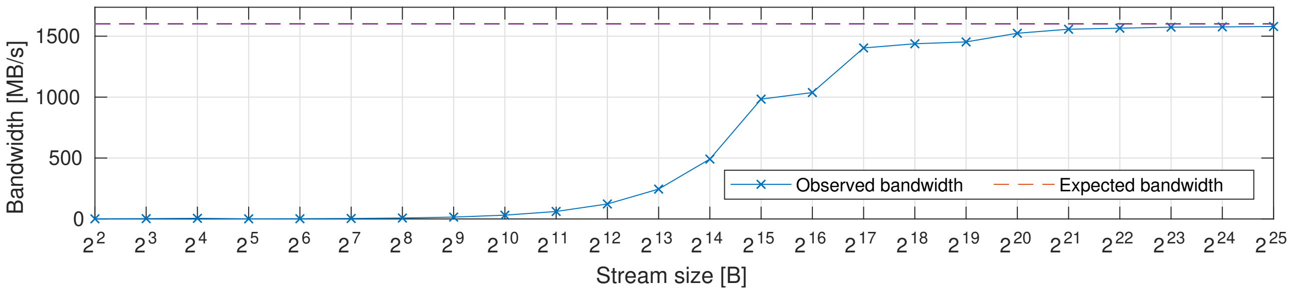


Figure 7: Observed duplex bandwidth for different stream sizes. The system was designed using a 64-bit F2S port and a loop-back FIFO, allowing to fully exploit the duplex bandwidth of the F2S port. The FPGA operation frequency is 100 MHz. Each data point represents the average of 200 transactions of the same size. The maximum standard deviation was 42.19 MB/s.

5.1 Experimental Results

The three systems to evaluate the bandwidth of the on-chip high-performance interfaces of the Cyclone V device were implemented using Intel Quartus Prime 18.1 with a FPGA operation frequency of 100 MHz. The MSGDMA was configured for a maximum burst size of sixteen, and a maximum stream size of 256 MB. The F2S port was tested for 32, 64, 128 and 256-bit configurations. The hardware requirements of the implemented systems are listed in Table 7, in Appendix A.

Table 5 summarizes the performance results obtained using all the three systems. The rows painted in yellow correspond to sub-optimal configurations whose bandwidth is degraded compared with the theoretically expected. The highest bandwidth (2779.05 MB/s) was achieved for HPS-to-FPGA transfers using all the 256 bit of the F2S port, which is compatible with the results obtained by Göbel *et al.*. Similarly to the Xilinx device, for configurations where the F2S interface is wider, the bandwidth is degraded. Furthermore, that effect is exaggerated by the lower operating frequency of the DRAM connected to the Cyclone V device comparing with the one connected to the ZYNQ-7000 device, which leads to a lower DRAM bandwidth. It is also worth noticing that the maximum bandwidth allowed for duplex transfers is lower than the one associated with single-sided transfers. This effect is also visible in the results regarding the Xilinx device and can be attributed to the entropy generated at the DRAM controller level when multiplexing the DDR ports to read and write data at data rates close to the DRAM's limit. This hypothesis is supported by the higher standard deviations associated with the results of experiments using those configurations, which indicate that they are not completely deterministic.

Concluding the analysis of the Cyclone V device family, a final scenario based on the system to evaluate the duplex bandwidth was considered. The F2S port was configured to use only 64 bit and the system was implemented targeting an operation frequency of 100 MHz. This experiment consisted of transferring streams of multiple sizes from the HPS to the FPGA and back to the HPS. Each stream size was tested 200 times, and the results were averaged. Figure 7 illustrates the results of this experiment. Results show that small stream sizes only achieve a fraction of the theoretical maximum bandwidth allowed by the F2S port. Only for stream sizes bigger than 512 KiB a real bandwidth of more than 90% of the theoretical maximum is achieved.

Table 5: Expected and observed bandwidth and respective system efficiency for several system configurations using F2S ports of the Cyclone V device. Each configuration was tested 200 times for 32 MiB data blocks. The FPGA operating frequency was 100 MHz. Note that H2F stands for HPS-to-FPGA, and F2H means FPGA-to-HPS.

	Channel Width [bit]	**Bandwidth** [MB/s]					**Efficiency** [%]
		Expected	**Observed**				
			Average	**Minimum**	**Maximum**	σ	
Duplex	**32**	800.00	799.59	799.04	799.91	0.1279	99.95
	64	1600.00	1579.80	1563.76	1583.54	3.1448	98.74
	128	3200.00	2019.26	1836.18	2023.30	13.9767	63.10
	256	3200.00	2037.16	1885.93	2039.91	13.1620	63.66
H2F	**32**	400.00	399.81	399.69	399.96	0.0299	99.95
	64	800.00	799.60	797.81	799.87	0.1862	99.95
	128	1600.00	1569.88	1561.47	1570.90	1.5062	98.12
	256	3200.00	2779.05	2736.46	2790.85	5.9443	86.85
F2H	**32**	400.00	399.81	399.70	399.95	0.0333	99.95
	64	800.00	799.59	797.83	799.87	0.2206	99.95
	128	1600.00	1576.78	1574.66	1579.18	1.8717	98.55
	256	3200.00	2464.66	2137.91	2473.06	32.3298	77.02

6 Conclusions

This work aimed at evaluating the performance of the on-chip high-performance interfaces of Xilinx and Intel's low-end FPGA-SoC device families (ZYNQ-7000 and Cyclone V, respectively). For achieving that goal, three systems were designed to fully exploit the duplex, PS-to-PL, and PL-to-PS bandwidths. Then, those systems were implemented and targeted on both Xilinx ZYNQ-7010 and Intel Cyclone V SE devices. Multiple implementation options were considered regarding the choice of DMA engines to use on each device and the software environment running on the hard CPUs of the FPGA-SoCs (bare-metal or Linux).

The obtained results were compared to the results obtained by Göbel *et al.* in a previous study, and also with the theoretical limits of the devices. The maximum observed data rates are similar to the ones observed by Göbel *et al.* Additionally, the results suggest that the bandwidth allowed by the on-chip high-performance channels of both devices is limited by the memory controller. Whenever getting close to the DDR maximum data rate, the bandwidth starts to degrade. Furthermore, the higher bandwidths achieved by single-sided transfers and the higher standard deviations associated with experiments using sub-optimal configurations suggest that there is some level of entropy at the memory controller level that affects the performance negatively.

The artifact produced by this work and its support documentation can be found at `https://github.com/joaomiguelvieira/FPGA_SoC_DMA_stress/`.

Acknowledgments

I would like to thank Professor Horácio Neto and Professor Rui Duarte for their support and guidance during the execution of this project.

I also thank Sahand Kashani for his support, namely the numerous hours spent on debugging the designs for the Intel FPGA-SoC device.

References

[1] Mohammadsadegh Sadri, Christian Weis, Norbert Wehn, and Luca Benini. Energy and performance exploration of accelerator coherency port using Xilinx ZYNQ. In *Proceedings of the 10th FPGAworld Conference*, pages 1–8, 2013.

[2] Valery Sklyarov, Iouliia Skliarova, João Paulo Sá da Silva, and Alexander Sudnitson. Analysis and Comparison of Attainable Hardware Acceleration in All Programmable Systems-on-Chip. In *DSD*, pages 345–352. IEEE Computer Society, 2015.

[3] Mohammad Tahghighi, Sharad Sinha, and Wei Zhang. Analytical delay model for CPU-FPGA data paths in programmable system-on-chip FPGA. In *International Symposium on Applied Reconfigurable Computing*, pages 159–170. Springer, 2016.

[4] Bo Joel Svensson. Exploring OpenCL Memory Throughput on the Zynq. Technical report, Technical Report, 2016.

[5] Matthias Göbel, Ahmed Elhossini, Chi Ching Chi, Mauricio Alvarez Mesa, and Ben H. H. Juurlink. A Quantitative Analysis of the Memory Architecture of FPGA-SoCs. In *ARC*, volume 10216 of *Lecture Notes in Computer Science*, pages 241–252, 2017.

[6] Digilent. Zybo. `https://reference.digilentinc.com/reference/programmable-logic/zybo/start`, accessed on May 17th, 2020.

[7] TerasIC. DE1-SoC Board. `https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=836`, accessed on May 17th, 2020.

[8] Xilinx. AXI DMA v7.1: LogiCORE IP Product Guide, PG021. `https://www.xilinx.com/support/documentation/ip_documentation/axi_dma/v7_1/pg021_axi_dma.pdf`, 2019.

[9] Xilinx. AXI Reference Guide, UG761 (v13.1). `https://www.xilinx.com/support/documentation/ip_documentation/ug761_axi_reference_guide.pdf`, 2011.

[10] Xilinx. Zynq-7000 All Programmable SoC: Technical Reference Manual, UG585 (v1.12.2). `https://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf`, 2015.

[11] Intel. Intel Avalon Specifications. `https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/manual/mnl_avalon_spec.pdf`, 2020.

[12] Sahand Kashani and René Beuchat. SoC-FPGA Design Guide, DE1-SoC Edition. `https://github.com/sahandKashani/SoC-FPGA-Design-Guide/blob/master/DE1_SoC/SoC-FPGA%20Design%20Guide/SoC-FPGA%20Design%20Guide%20%5BDE1-SoC%20Edition%5D.pdf`, accessed on May 17th, 2020.

A Hardware Resources

Table 6: Hardware requirements of the systems to assess the on-chip high-performance interfaces of the Xilinx ZYNQ-7010 device.

	Channel Width [bit/channel]	**Hardware Resources**							
		LUTs (17600)		**FFs (35200)**		**BRAMs (60)**		**DSPs (80)**	
Duplex	**32**	8581	(48.76 %)	11702	(33.24 %)	14	(23.33 %)	0	(0.00 %)
	64	10045	(57.07 %)	13594	(38.62 %)	22	(36.67 %)	0	(0.00 %)
PS to PL	**32**	2623	(14.90 %)	3769	(10.71 %)	4	(6.67 %)	0	(0.00 %)
	64	2656	(15.09 %)	3697	(10.50 %)	2	(3.33 %)	0	(0.00 %)
PL to PS	**32**	5060	(28.75 %)	6685	(18.99 %)	4	(6.67 %)	0	(0.00 %)
	64	5671	(32.22 %)	7421	(21.08 %)	6	(10.00 %)	0	(0.00 %)

Table 7: Hardware requirements of the systems to assess the on-chip high-performance interfaces of the Intel Cyclone V SE device.

	Channel Width [bit]	**Hardware Resources**						
		ALMs		**Registers**	**BRAMs**		**DSPs**	
Duplex	**32**	1036	(3.23 %)	1538	14	(3.53 %)	0	(0.00 %)
	64	1033	(3.22 %)	1536	20	(5.04 %)	0	(0.00 %)
	128	1064	(3.32 %)	1537	32	(8.06 %)	0	(0.00 %)
	256	1105	(3.45 %)	1535	58	(14.61 %)	0	(0.00 %)
HPS to FPGA	**32**	803	(2.50 %)	1270	3	(0.76 %)	0	(0.00 %)
	64	817	(2.55 %)	1261	3	(0.76 %)	0	(0.00 %)
	128	800	(2.49 %)	1255	3	(0.76 %)	0	(0.00 %)
	256	818	(2.55 %)	1262	3	(0.76 %)	0	(0.00 %)
FPGA to HPS	**32**	789	(2.46 %)	1210	7	(1.76 %)	0	(0.00 %)
	64	780	(2.43 %)	1214	10	(2.52 %)	0	(0.00 %)
	128	798	(2.49 %)	1209	16	(4.03 %)	0	(0.00 %)
	256	856	(2.67 %)	1210	29	(7.30 %)	0	(0.00 %)