StreamIF - High Performance Memory Interface for Reconfigurable Operating Systems on Xilinx ZedBoard Platform

Taner Guven
Dept. of Computer Engineering
Kocaeli University
Kocaeli, Turkey

Suhap Sahin
Dept. of Computer Engineering
Kocaeli University
Kocaeli, Turkey

ABSTRACT
Today, FPGA-based Hardware/Software Co-design applications are frequently used in many areas such as high performance computing, video processing, automation, automotive, and communications. Since there is not any standardized operating system used for FPGA applications, it is usually required to start developing from scratch for both software and hardware. Although this need is met partly by the Reconfigurable Operating System studies, it has not been reached yet to the level of performance of bare-metal applications. In this article, the design and implementation of the memory interface, which was developed for high performance data transfer between software and hardware in reconfigurable operating systems, was described. The memory interface (StreamIF), which was developed as open source, was verified with the ReconOS operating system in "Xilinx Zedboard Zynq-7000 SoC Development Board".

General Terms
Reconfigurable Operating Systems, Hardware/Software Co-design

Keywords
FPGA, Memory Interface, Zynq-7000, AXI Stream, AXI DMA

1. INTRODUCTION
Today, the hardware development methods with FPGA widely used have been fell behind the developed software technology. In FPGA programming, there is not any standardized and widely used method in the subjects such as inter-process communication, memory management, input/output management provided by the operating system. It is usually required to start developing from scratch in order to develop a new application with FPGA. Many studies have been carried out related to Reconfigurable Operating System (ROS) in order to meet this deficit [1] [2] [3] [4].

ROS works facilitate the development process of Hardware/Software Co-design applications by providing a standardized interface for accessing the operating system services such as I/O management, memory management, resource management, inter-process communication via software and hardware. With the "dynamic partial reconfiguration" feature provided by modern FPGAs, the FPGA content can be changed during the run-time and used as time-shared. With identification of FPGA modules by the ROS's as the "hardware process" or "hardware thread", a process management can be done during the run-time similarly to the software.

In real-time signal processing applications such as video processing, high data transfer rates are needed between software and hardware. When their data transfer rates are compared, it can be reached to the rates of around 200-240mb/s with the applications developed by using the ROS [5], it can be reached to the rates of around 2900MB/s with the bare-metal applications [6], in which an operating system is not used.

The target of our study is to develop a "free and open source" memory interface module which can be used in "Reconfigurable OS" with the data transfer performance level between hardware and software in bare-metal applications. The developed memory interface (StreamIF) has been verified with the ReconOS operating system in the "Xilinx ZedBoard Zynq-7000 SoC Development Board"[7]. ReconOS is a "free and open-source" reconfigurable operating system which supports the Xilinx ZedBoard kit.

In this article, internal structure of StreamIF, its integration with Zynq-7000 architecture and its use with ReconOS was described. StreamIF was compared with the theoretical limits of the memory interface and platform used in ReconOS. In addition, a performance comparison was made between the study and a high-performance application which does not use an operating system.

2. RECONOS MEMORY INTERFACE
In ReconOS, there are two interfaces as OSIF and MEMIF, to maintain a communication between hardware thread (HWT) and software thread (SWT) [8]. OSIF is the system call interface on the hardware side of the operating system and it includes the inter-process communication functions used by the HWT. On the other hand, MEMIF is the interface which enables HWTs to access to the memory.

The "HWT memory access model" used in ReconOS was shown in Fig 1. The Memory Subsystem is connected to the processing system (PS) via the AXI_ACP interface. The AXI_ACP interface accesses the DDR RAM via the L2 cache. Memory access made by using the L2 cache provides an advantage in terms of reducing the delay in the transfer of small and medium size data between the CPU and the FPGA. Although, it may cause problems such as cache thrashing in applications requiring high bandwidth such as video processing, or in applications which has new data transmit continuously.

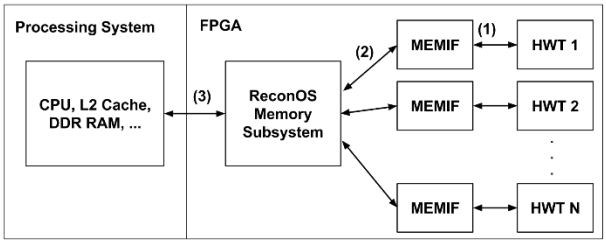


Fig 1: HWT memory access model in ReconOS

The memory subsystem supports read/write bandwidth of 400 MB/s theoretically since it operates as half-duplex in 32-bit wide and 100MHz frequency. In the performance tests, that we made, this value was measured around 236 MB/s. The bandwidth provided by the memory subsystem is quite lower compared to the 4264 MB/s DDR RAM bandwidth offered by the Zynq-7000 architecture.

In Table 1, the features of AXI_ACP and AXI_HP interfaces that can be used for PS communication with the FPGA for the Zynq-7000 platform and the features of DDR RAM were given [9].

Table 1. AXI_ACP, AXI_HP, DDR RAM features on Zynq-7000 platform

	AXI_ACP	AXI_HP	DDR
Bus Width (Bits)	64	64	32
Clock (MHz)	150	150	1066
Read Bandwidth (MB/s)	1200	1200	4264
Write Bandwidth (MB/s)	1200	1200	4264
R+W Bandwidth (MB/S)	2400	2400	4264
Number of Interfaces	1	4	1
Total Bandwidth (MB/S)	2400	9600	4264
ReconOS Memory Subsystem Limit *	**400**	**unused**	**400**

*max bandwidth for 100 MHz 32 bit simplex data transfer

The memory subsystem provided by ReconOS provides a very easy and fast developing environment to the programmer for small and medium size data. Although, alternative methods are needed for problems requiring high memory bandwidth.

3. PROPOSED MEMORY INTERFACE MODEL

StreamIF is optimized for stream-type data and developed to meet the memory interface requirement which can use the main memory bandwidth with high efficiency. In StreamIF's connection with Processing System, the AXI_HP interface which can correspond the main memory bandwidth, was preferred. Therefore, it was aimed to achieve the highest bandwidth that can be reached with a design in which all 4 AXI_HP ports on the platform are used.

In Fig 2 the memory access model, which has been created by adding the StreamIF to ReconOS was shown. StreamIF has been added as the second memory interface without changing the existing structure of ReconOS shown in Fig 1. In this model, it was aimed to enable the HWTs to use MEMIF and StreamIF according to their needs.

In the connection of Processing System and StreamIF, four data buses with 64 bit wide 100 MHz frequency. In this data bus, in which AXI4 [10] protocol is used, theoretically a total of 6400MB/s bandwidth is supported including 3200MB/s read and 3200MB/s write.

Two data buses have been used to enable simultaneous reading and writing to be made between StreamIF and HWTs. By running the buses at 32 bit wide and 100 MHz frequency, theoretically a total of 800 MB/s bandwidth is supported for each HWT, including 400 MB/s read and 400 MB/s write.

The AXI4 Stream [11] protocol is preferred between StreamIF and HWTs. It is possible to make conversion from the AXI4 Stream protocol, the AXI4 protocol used in connection to the Processing System by using fewer resources. In addition, the AXI4 Stream protocol facilitates the integration of many tools and libraries into the system, including the Xilinx High Level Synthesis.

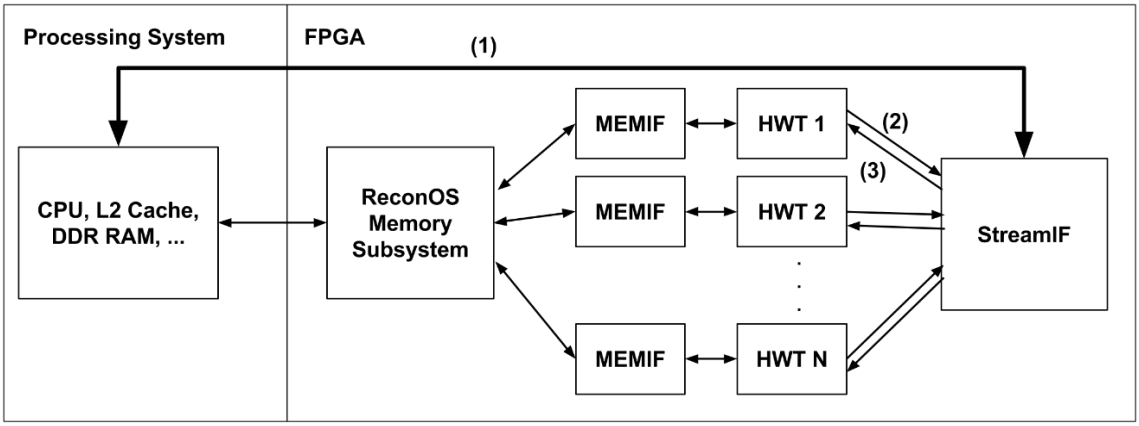


Fig 2: HWT memory access model with StreamIF

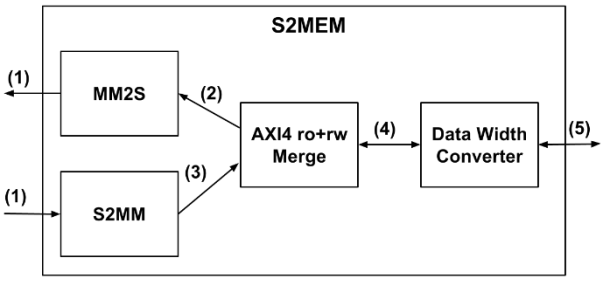
4. INTERNAL STRUCTURE OF STREAMIF

StreamIF has been designed to increase the number of connections according to the number of HWTs. Fig 5 shows the internal structure and connections of the StreamIF module in case 8 HWT is used. StreamIF consists of a control module and S2MEM modules in the same number of HWTs. The S2MEM module enables HWT to access the main memory. The "StreamIF Control" module enables to manage the S2MEM module through the operating system.

The connection between the S2MEM and the PS has been made so that two S2MEMs and an AXI_HP port will match since there are 4 AXI_HP ports in the platform. For the protocol conversion and multiplexing between StreamIF and Zynq-7000, the "Xilinx AXI Interconnect" [12] module, which is available as a standard in the Xilinx development tools, has been used.

A 64-bit data bus has been used between the S2MEM module and the AXI Interconnect to enable the AXI Interconnect to operate efficiently in 64-bit mode. Since a 32-bit data bus has been used between S2MEM and HWT, then AXI Interconnect can be used with full efficiency in case at least 2 HWTs are active. At least 8 HWTs must be active in order to achieve full performance throughout the system.

The internal structure of the S2MEM module is shown in Fig 3. S2MEM includes Direct Memory Access modules named MM2S and S2MM which can be programmed through the control module. MM2S reads the data from the memory and sends it to the HWT, on the other hand the S2MM writes the data, which comes from the HWT, to the memory.



(1) 32 bit AXI4 Stream
(2) 32 bit read-only AXI4
(3) 32 bit write-only AXI4
(4) 32 bit AXI4
(5) 64 bit AXI4

Fig 3: Internal structure of S2MEM module

The MM2S module reads the data with the AXI4 protocol and provides output with the AXI4 Stream protocol. In Fig 4 the internal structure of the MM2S module was shown. AXI4-Reader and AXI4-Stream-Writer are directly connected to each other. The data read from the AXI4 port is transferred to the AXI4-Stream port in the same clock (without any delay). The MM2S-FSM ensures that the reader and writer modules operate in a coordinated manner.

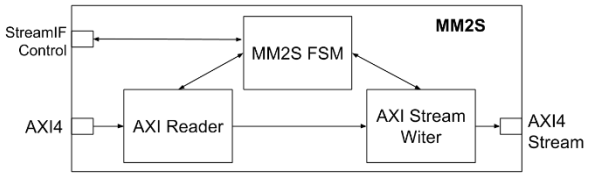
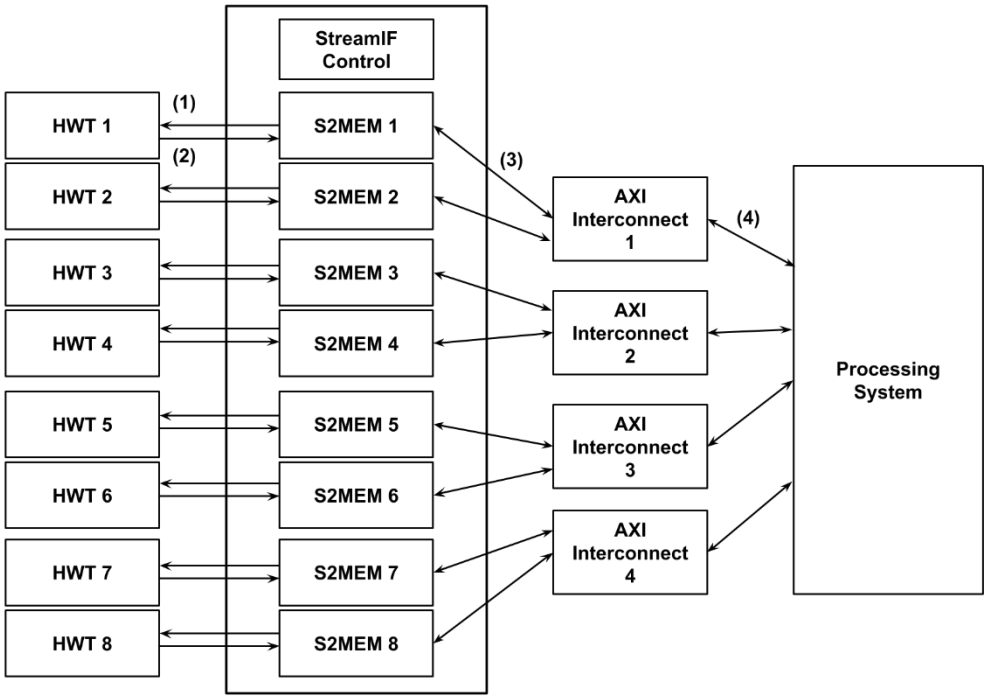


Fig 4: Internal structure of MM2S module



(1) 32 bit 100 MHz AXI4 Stream
(2) 32 bit 100 MHz AXI4 Stream
(3) 64 bit 100 MHz AXI4
(4) 64 bit 100 MHz AXI4

Fig 5: Internal structure and connections of StreamIF

The duty of the S2MM module is to read the data in the AXI4-Stream protocol and convert it to the AXI4 protocol, contrary to MM2S. In Fig 6 the internal structure of the S2MM module was shown. The internal structure of S2MM is similar to that of MM2S, and the working logic is the same.

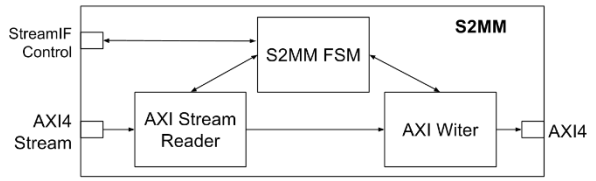


Fig 6: Internal structure of S2MM module

The Data Width Converter module in Fig 3 has been used to solve the data bus width difference problem between HWT and AXI_HP port. This problem can be solved also by using the data width converter inside the "AXI Interconnect" in Xilinx development tools. Although, since the "Xilinx Data Width Converter" has a high resource utilization and it is not sufficiently adaptable to our problem, a data width converter specific to the problem have been developed in our study. The resource utilization comparison of data with converters is given in the Table 2.

Table 2. Data width converter resource utilization comparison

	Xilinx Data Width Converter	StreamIF Data Width Converter
Slice	344	46
Slice Reg	887	116
LUT	738	115
BRAM	3	2

In Fig 7, the internal structure of the developed data width converter has been showed. This module bit makes conversion in AXI protocol from 32 bit-width to 64 bit-width. Due to bit conversion, 256 x 32 bit is transferred in 32-bit interface in one transaction, while 128 x 64 bits are transferred in a 64 bit interface. A 512-deep 64-bit BRAM based FIFO has been used for synchronization between the interfaces.

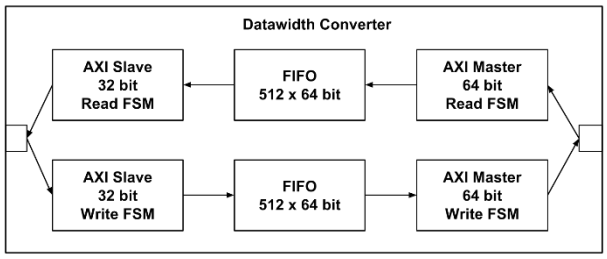


Fig 7: Internal structure of data width converter

5. TEST RESULTS

The performance tests of the system were made with ReconOS on Xilinx ZedBoard. In performance tests, ReconOS Memory Subsystem and our study (StreamIF) were compared in write-only, read-only and read+write categories. In the tests, it was aimed to measure the approximate resource utilization of the operating system by using programs which do not include complex calculations but focus only on read/write operations.

The read+write test was performed with HWT which reads 4 MB data from one address of RAM and writes it down to another address. In Table 3 the comparison of the total bandwidth usage in the read+write test was given.

Table 3. read+write bandwidth utilization

Hardware Thread Count	MEMIF R+W Speed (MB/s)	StreamIF R+W Speed (MB/s)
1	224.56	622.45
2	235.12	1241.06
3	234.92	1843.05
4	235.34	2432.96
5	235.31	2844.05
6	235.22	2908.33
7	235.23	2989.10
8	235.46	3039.14

In the read+write test, the ReconOS Memory Subsystem was able to reach the speed rates up to 236 MB/s without being affected by the HWT count. When StreamIF was used, the values between 622 MB/s and 3039 MB/s were observed depending on the number of HWTs.

DDR controller efficiency value was stated by Xilinx around 75% h for the Zynq-7000 platform [13]. In practice, In Xilinx's "High-Performance Video Systems" [6] 70% memory bandwidth utilization value was achieved. In our study, we obtained 71.2% efficiency by using around 3039 MB/s from 4264 MB/s main memory bandwidth in case 8 HWT was active.

In Table 4, the performance comparison of read-only and write only tests was given. The read-only test has been performed by reading a 32 bit integer array of 1048576 elements from RAM and XOR operation on its elements. On the other hand, the write only test has been performed by saving the output of a counter to RAM.

Table 4. Read-only and write-only bandwidth utilization

	MEMIF		StreamIF	
HWT Count	Read Speed (MB/s)	Write Speed (MB/s)	Read Speed (MB/s)	Write Speed (MB/s)
1	218.13	254.09	331.68	340.11
2	230.67	255.63	650.57	652.52
3	229.13	256.06	969.42	971.51
4	229.62	256.52	1282.69	1290.32
5	229.43	257.10	1608.75	1624.06
6	229.27	257.11	1905.71	1925.54
7	228.89	257.09	2219.13	2244.00
8	228.56	257.09	2524.19	2547.58

The resource utilization in the case of using 8 HWTs was given in the Table 5. With the addition of StreamIF into ReconOS, the resource utilization in the Slice category increased by about 20% and risen to 44%.

Table 5. Resource utilization when using 8 HWTs

	MEMIF Only	**MEMIF + StreamIF**
Slice	4832 (%36.3)	5794 (%43.6)
Slice Reg	6164 (%5.8)	11129 (%10.5)
LUT	12276 (%23.1)	13440 (%25.3)
BRAM	8 (%5.7)	16 (%11.4)

6. CONCLUSION

In this article, the design and implementation of a high-performance memory interface developed for reconfigurable operating systems was presented. According to the performance results obtained in the tests, approximately 12 times higher memory bandwidth was obtained with the addition of StreamIF into ReconOS, it was achieved to use approximately 71% of the theoretical bandwidth value of RAM in practice. In comparison with the obtained 12-times increase in performance, the resource utilization was increased only 20%.

For further study, it is aimed to add memory protection mechanism to StreamIF for memory isolation between HWTs and processes. Furthermore, it is possible to reduce memory bandwidth usage by adding direct data transfer support between HWTs.

StreamIF's source codes have been published under the LGPLv2.1 license from the [14] web address. The test codes and outputs used in this article have been published on the [15] web address.

7. REFERENCES

[1] So, H.K.H. and Brodersen, R., 2008. A unified hardware/software runtime environment for FPGA-based reconfigurable computers using BORPH. ACM Transactions on Embedded Computing Systems (TECS), 7(2), p.14.

[2] Andrews, D., Sass, R., Anderson, E., Agron, J., Peck, W., Stevens, J., Baijot, F. and Komp, E., 2008. Achieving programming model abstractions for reconfigurable computing. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 16(1), pp.34-44.

[3] Jozwik, K., Honda, S., Edahiro, M., Tomiyama, H. and Takada, H., 2013. Rainbow: An operating system for software-hardware multitasking on dynamically partially reconfigurable fpgas. International Journal of Reconfigurable Computing, 2013, p.5.

[4] Agne, A., Happe, M., Keller, A., Lubbers, E., Plattner, B., Platzner, M. and Plessl, C., 2014. ReconOS: An operating system approach for reconfigurable computing. IEEE Micro, 34(1), pp.60-71.

[5] Wang, Y., Zhou, X., Wang, L., Yan, J., Luk, W., Peng, C. and Tong, J., 2013. Spread: A streaming-based partially reconfigurable architecture and programming model. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 21(12), pp.2179-2192.

[6] Lucero, J. and Arbel, Y., 2012. Designing High-Performance Video Systems with the Zynq-7000 All Programmable SoC.

[7] ZedBoard. [Online]. Available: https://www.xilinx.com/products/boards-and-kits/1-8dyf-11.html, accessed May. 11, 2015.

[8] Architecture of ReconOS, [Online]. Available: http://www.reconos.de/documentation/architecture/, accessed May. 11, 2015.

[9] Crockett, L.H., Elliot, R.A., Enderwitz, M.A. and Stewart, R.W., 2014. The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc. Strathclyde Academic Media.

[10] AMBA AXI and ACE Protocol Specification. [Online]. Available: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ihi0022e/index.html, accessed Mar. 15, 2016.

[11] AMBA AXI4-Stream Protocol Specification. [Online]. Available: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ihi0051a/index.html, accessed Mar. 15, 2016.

[12] AXI Interconnect (v1.06.a). [Online]. Available: https://www.xilinx.com/support/documentation/ip_documentation/axi_interconnect/v1_06_a/ds768_axi_interconnect.pdf, accessed Sep. 28, 2016

[13] Zynq-7000 All Programmable SoC Technical Reference Manual. [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf, accessed Nov. 9, 2016.

[14] StreamIF source codes. [Online]. Available: https://github.com/tanerguven/StreamIF, accessed May. 9, 2017

[15] StreamIF ReconOS test codes. [Online]. Available: https://github.com/tanerguven/ReconOS-StreamIF-tests/tree/201705-StreamIF-Article, accessed May. 9, 2017