Göbel, M.; Elhossini, A.; Chi, C. C.; Álvarez-Mesa, M.; Juurlink, B.

A Quantitative Analysis of the Memory Architecture of FPGA-SoCs

The final authenticated version is available online at
https://doi.org/10.1007/978-3-319-56258-2_21.

Göbel, M., Elhossini, A., Chi, C. C., Álvarez-Mesa, M., & Juurlink, B. (2017). A Quantitative Analysis of the Memory Architecture of FPGA-SoCs. In: Applied reconfigurable computing : 13th International Symposium, ARC 2017 (Lecture Notes in Computer Science, vol. 10216), pp. 241–252. Springer International Publishing. https://doi.org/10.1007/978-3-319-56258-2_21

A Quantitative Analysis of the Memory Architecture of FPGA-SoCs

Matthias Göbel[1(*)], Ahmed Elhossini[1], Chi Ching Chi[2],
Mauricio Alvarez-Mesa[2], and Ben Juurlink[1]

[1] Embedded Systems Architecture, Technische Universität Berlin, Berlin, Germany
{m.goebel,ahmed.elhossini,b.juurlink}@tu-berlin.de
[2] Spin Digital Video Technologies GmbH, Berlin, Germany
{chi,mauricio}@spin-digital.com

Abstract. In recent years, so called *FPGA-SoCs* have been introduced by Intel (formerly Altera) and Xilinx. These devices combine multi-core processors with programmable logic. This paper analyzes the various memory and communication interconnects found in actual devices, particularly the Zynq-7020 and Zynq-7045 from Xilinx and the Cyclone V SE SoC from Intel. Issues such as different access patterns, cache coherence and full-duplex communication are analyzed, for both generic accesses as well as for a real workload from the field of video coding. Furthermore, the paper shows that by carefully choosing the memory interconnect networks as well as the software interface, high-speed memory access can be achieved for various scenarios.

1 Introduction

HW/SW-codesign is a common approach applied in domains where neither pure hardware nor pure software implementations offer a satisfying solution. It combines the advantages of both hardware and software and therefore delivers an elaborated solution to a given problem. FPGA manufacturers such as Xilinx and Intel are offering devices, often called *FPGA-SoCs*, that combine an FPGA logic fabric and a dedicated processor, which in the end allows for a significant performance gain when using HW/SW-codesign compared to pure software solutions.

In order to achieve high speedup, it is clearly important to achieve high performance of both the hardware and the software. However, without having sufficient memory bandwidth it is not possible to unleash the full potential of such a solution. In fact, the memory bandwidth often poses the bottleneck in HW/SW-codesigns and therefore limits the overall performance: While it is possible to achieve a very high throughput in an FPGA, the memory interface is in many cases not able to provide input and store output data fast enough [1,2]. Therefore, many research papers are only presenting the throughput inside the FPGA while disregarding the memory bandwidth [3,4]. For this reason, this work presents an analysis of the memory architecture of FPGA-SoCs.

Two representative low-cost FPGA-SoCs have been chosen for the analysis, particularly the *Zynq-7020* from Xilinx and the *Cyclone V SE* SoC from Intel.

Furthermore, the same benchmarks have been performed on the *Zynq-7045* from Xilinx to show the memory bandwidth of a high-performance FPGA-SoC. These results have also been compared to a system using a configurable soft-core memory controller from Xilinx. This allows for a comparison of the memory bandwidth of FPGA-SoCs with soft-core SoCs using Xilinx's *Microblaze* or Intel's *Nios II*. The best configurations for all these devices are discussed and their respective strengths are highlighted.

The main contribution of this paper is the evaluation of the memory subsystems of the Zynq-7000 SoC from Xilinx and the Cyclone V SoC from Intel, taking into account all of the following:

1. Memory access from software as well as from hardware
2. Coherent as well as non-coherent access
3. Independent read and write transactions
4. Coupling of multiple memory ports
5. Fine-grained, two-dimensional transactions that are often found in video coding and image processing kernels
6. Evaluation of the available memory bandwidth for H.265/HEVC motion compensation as a representative for such video coding kernels

The paper is structured as follows: First, some related work is presented in Sect. 2 to give an overview of the current state-of-the-art. Then, in Sect. 3, a short introduction to the FPGA-SoCs from Intel and Xilinx is given with a focus on their memory interface. This is followed in Sect. 4 by a description of the implemented memory engines that are used to measure the bandwidth under various circumstances. In Sect. 5, the Zynq-7020 and the Cyclone V SE SoC are evaluated and compared, followed by an analysis of the Zynq-7045 and Xilinx's soft-core memory controller. Finally, the paper is concluded in Sect. 6.

2 Related Work

Some other work already evaluated the memory bandwidth of FPGA-SoCs. First results are given by Sadri et al. [5]. They analyzed the memory interfaces of the Zynq-7020 with a focus on the *Accelerator Coherency Port* (ACP), which allows coherent access from IP cores implemented in logic to main memory. The results show that it is possible to achieve a full-duplex throughput of up to 1.7 GB/s when using a single port between memory and programmable logic, with the IP core running at a fixed frequency of 125 MHz.

Sklyarov et al. [6] also evaluated the Zynq-7020. Although the maximum bandwidth at the chosen frequency of 100 MHz is not given explicitly, it can be derived from the results that the achieved maximum bandwidth is significantly lower than the theoretical maximum (e.g. 284 MB/s for a 64-bit port when reading and writing 32 KB instead of the theoretically possible 800 MB/s).

Furthermore, Tahghighi et al. [7] present a mathematical model that allows to estimate the latency of a memory access from the programmable logic. While the model covers several parameters, it is currently limited to the Zynq-7000. It also does not give an overview of the available memory bandwidth for different access patterns. Similar to [5], it does not cover the combination of multiple ports to increase the overall memory bandwidth.

Although these papers provide valuable information, several of our questions remain unanswered. For instance, the combination of multiple ports yields a significant increase in bandwidth thus expanding the field of applications suitable for FPGA-SoCs to a broader range. While this is analyzed in [6], their results are surprisingly low. In comparison, our results show a significantly higher bandwidth when combining multiple ports. Furthermore, to the best of our knowledge, our work is the first to include multiple devices that cover a large part of the market (Xilinx's Zynq-7020 and Zynq-7045 + Intel's Cyclone V SE SoC + Xilinx's Microblaze) while all the related papers only use the Zynq-7020 for their evaluations thus limiting their impact.

3 FPGA-SoCs

FPGA-SoCs are devices that contain a dedicated hard-core processor with various peripherals and programmable logic. Both components are located on the same chip, which allows them to be tightly coupled. Such devices are offered by Xilinx [8] and Intel [9]. Both combine a 32-bit dual-core ARM Cortex-A9 based CPU with programmable logic. This CPU uses the ARMv7-A architecture and support NEON SIMD instructions. A two-level cache hierarchy is available that provides 32 KB of L1 per core and a shared 512 KB L2 cache.

Xilinx offers the Zynq-7000 family of so-called *All-programmable SoCs* while Intel offers SoCs as part of their Cyclone, Stratix and Arria product lines to cover the whole market. Both vendors have already announced successors to their current FPGA-SoCs, featuring a 64-bit quad-core ARM Cortex-A53 CPU and more logic resources. However, as they were not publicly available at the time of this work, they could not be included.

While Xilinx devices use only support the ARM AXI standard, Intel supports AXI as well as their own Avalon standard. For the sake of comparison, only the AXI mode of Intel's devices was taken into account. Both vendors offer a variety of master and slave ports suitable for different applications. As the master ports (i.e. the CPU is the master) cannot be used directly to access the DDR memory from the programmable logic, these ports will not be discussed in this work.

Xilinx's Zynq-7000 devices offer the following ports for the programmable logic to access memory:

1. **General-purpose** (GP) ports
   These two ports have a fixed width of 32 bits and no internal buffers, making them a good choice for low-throughput applications.
2. **High-performance** (HP) ports
   Four slave ports with widths of either 32 or 64 bits with built-in FIFOs are available for high-throughput applications.
3. **Accelerator Coherency Port** (ACP)
   This additional 64-bit port resembles the HP ports. However, the ACP allows cache-coherent access to the memory.

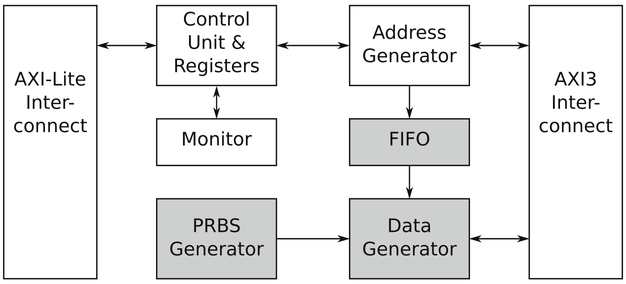


Fig. 1. The engines that are used to perform one- or two-dimensional accesses to main memory. A register-based AXI-Lite interface for control tasks and a full-scaled AXI master interface for data transfer connect the engine to the CPU and the main memory. Note that the gray blocks are only required for the write engine.

For Intel's SoCs, the layout of the ports for accessing memory is as follows:

1. **FPGA-to-HPS** (F2H) port
   This port has a configurable width of 32, 64 or 128 bits.
2. **FPGA-to-SDRAM** (F2S) port
   Instead of offering four ports like the Zynq's HP ports, the Intel SoCs have one port which is directly connected to the memory controller. This port, however, can be split into up to three independent AXI ports with a combined port width of up to 256 bits (e.g. 1×256-bit or 1×128-bit + 2×64-bit).
3. **Accelerator Coherency Port** (ACP)
   This port matches the ACP of the Zynq regarding DDR memory access.

4 Architecture of Memory Engines

In this section, the designs and implementations of the so-called *memory engines* are presented briefly. These engines allow to gain the required insights into the potential bandwidth of the different ports. They are designed to support one- and two-dimensional access to memory with a fixed stride, as well as trace-based inputs, i.e. a list of specific memory transactions. As this work focuses on high-throughput applications, the GP ports of the Zynq-7000 and the F2H port of Intel's SoC are not evaluated.

Figure 1 shows the general structure of the *Write Engine* that is used to determine the achievable write bandwidth for different scenarios. It has two different AXI interfaces: a full-scale AXI master interface for the actual memory access connected to one of the ports mentioned in Sect. 3 and a register-based AXI-Lite interface for control and configuration purposes. The latter is connected to the CPU using dedicated AXI ports that are not suitable for memory access. While Xilinx and Intel offer IP cores supporting AXI4, their FPGA-SoCs only support AXI3 for memory access. Therefore, the maximum number of bursts in one request is 16. By using the control interface, the specific scenario in terms of height and width of the access as well as the stride for two-dimensional access, i.e. the offset between two bytes in the same column, can be controlled.

The parameters stored in these registers are used by a *Control Unit*, which splits the two-dimensional block into one-dimensional transactions if necessary. These requests are afterwards converted into AXI transactions by an *Address Generator*. This unit is connected to the address lines of the AXI interface and drives the required signals. In addition, it deals with alignment issues. The requests are buffered in a FIFO from which they are read by a *Data Generator*. It writes the requested amount of data from a *Pseudo-Random Binary Sequence* (PRBS) generator to main memory.

To accurately measure the throughput of each operation, a *Monitor* has been added that measures the number of cycles the operation takes. It communicates with the CPU by using the register interface.

The implemented *Read Engine* for reading data from main memory has a very similar structure. However, as no data has to be generated and written for reading data, the corresponding generator and the FIFO are not required in this case.

5 Experimental Design and Performance Analysis

The implemented read and write engines have been used to evaluate the bandwidth of the interconnect ports and the memory system of the chosen FPGA-SoCs. In particular, two different benchmarks have been designed for this purpose:

1. A synthetic benchmark for one- or two-dimensional transactions. Performing a two-dimensional transaction can be understood as reading or writing a block of data (e.g. a part of an image) from/to memory with each row of the block consisting of one or multiple one-dimensional transactions.
2. A trace-based benchmark that simulates the memory transactions that are performed during H.265/HEVC motion compensation.

While the first benchmark gives an overview of the bandwidth that can be expected for a given width and height, the latter allows to measure the bandwidth for a real-world scenario with a mix of different block sizes. In this section, a comparison of the Zynq-7020 and the Cyclone V SoC will be discussed, as these are two chips in the same price segment. Later, the same benchmarks will be used to evaluate a high-performance FPGA-SoC, the Zynq-7045, in order to show the difference between low-cost FPGA-SoCs and high-performance FPGA-SoCs. Finally, a comparison to a system which uses Xilinx's soft-core memory controller instead of the hard-core memory controller of an FPGA-SoC will be presented. This allows comparing the bandwidth of the memory controller of an FPGA-SoC with that of a soft-core SoC such as Xilinx's Microblaze or Intel's Nios-II running on an FPGA.

All the benchmarks used in this work are optimized for high bandwidth. As a result, the highest possible number of data beats per burst is used.

5.1 Synthetic Benchmark

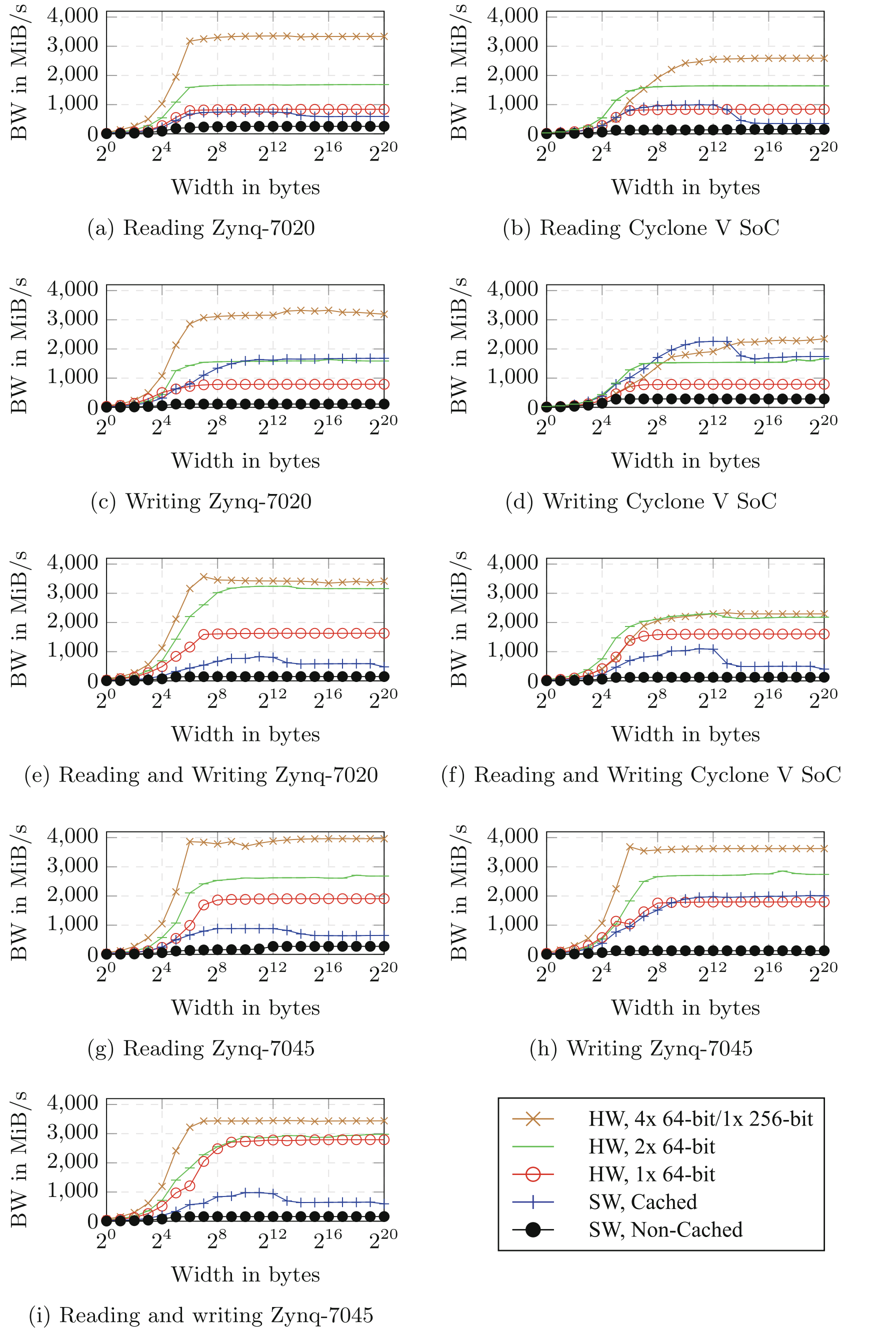
Cyclone V SoC and Zynq-7020. The experiments in this part have been performed using the *DE1-SoC Board* from Terasic that features Intel's Cyclone V SoC and the *Zedboard* from Digilent with Xilinx's Zynq-7020. The bandwidth is given in MiB/s, i.e. $2^{20}\,bytes/s$, and not in $10^{6}\,bytes/s$.

In order to get an overview of the achievable throughput for accessing different patterns in main memory, a synthetic benchmark has been used. It takes the width and height of the block being processed as well as the stride as parameters. The analyzed configurations include cached and non-cached software implementations as well as hardware implementations with different number of HP ports (Xilinx) or different widths of the F2S port (Intel) and with the ACP.

To have a reasonable baseline, the software implementations are NEON-accelerated, i.e. they use SIMD memory instructions to maximize the throughput. The non-ACP hardware implementations have been performed using a fixed frequency of 110 MHz for both the memory engine and the AXI bus, while the ACP implementation uses a frequency of 100 MHz. These are the maximum frequencies, i.e. the highest frequencies for which the memory engines could be placed and routed on all devices. The CPU on the Intel device is running at 800 MHz and also uses 800 MT/s for the memory controller. Xilinx uses a CPU with a frequency of 666 MHz, but 1066 MT/s to access the DDR memory. Due to the different memory data rates, the theoretical maximum bandwidth for DDR memory access is higher for the Zynq-7020 (4066 MiB/s) than for the Cyclone V SoC (3052 MiB/s). For all hardware experiments, the memory controller has been configured to prioritize the programmable logic memory ports and therefore minimize the impact of parallel memory accesses from software.

Figure 2(a)-(f) shows the results for the software and the non-ACP hardware scenarios. In this figure, a fixed stride of 1 MiB and a fixed height of 50 rows have been used while the width in bytes is the variable parameter with a range from 1 byte to 1 MiB. The choice of a height of 50 rows has been made as heights in this range are found quite often in video coding applications, an important domain when analyzing two-dimensional memory accesses. An example is the block structure of HEVC/H.265 [10]. A fixed stride of 1 MiB has been used as the stride must be larger or equal to the width. Thus, this choice allows for evaluating different memory accesses with a width of up to 1 MiB while using the same stride. Due to the choices of height and stride, this can either be interpreted as a single two-dimensional access with a height of 50 and a stride of 1 MiB or as 50 one-dimensional accesses with a fixed distance of 1 MiB between them. Therefore, it provides information for one- as well as two-dimensional access.

For reading, the non-cached SW baseline has the lowest throughput for both devices with a maximum bandwidth of 256 MiB/s on the Zynq-7020 and 150 MiB/s on the Cyclone V SoC. On the other hand, for the cached SW baseline, the Intel device has a significantly higher bandwidth of up to 996 MiB/s compared to a maximum of 751 MiB/s for its Xilinx counterpart. These differences are probably caused by the lower frequency of the Xilinx CPU and therefore of the caches. However, starting at around 16 KiB, i.e. the width where the 512 KiB L2



(a) Reading Zynq-7020

(b) Reading Cyclone V SoC

(c) Writing Zynq-7020

(d) Writing Cyclone V SoC

(e) Reading and Writing Zynq-7020

(f) Reading and Writing Cyclone V SoC

(g) Reading Zynq-7045

(h) Writing Zynq-7045

(i) Reading and writing Zynq-7045

Fig. 2. The bandwidth (BW) for a fixed stride of 1 MiB and a height of 50 rows. The HW implementations are running at 110 MHz (Zynq-7020 and Cyclone V SoC) and 214 MHz (Zynq-7045 4x/2x) or 250 MHz (Zynq-7045 1x). The CPUs are running at 666 MHz (Zynq-7020) or 800 MHz (Zynq-7045 and Cyclone V SoC). Note that for the combined read and write transactions the added bandwidth for reading and writing is given.

Cache can no longer hold the entire 50 rows, the Zynq-7020 again outperforms it counterpart. The stride of 1 MiB induces several cache misses in this case, which allows for comparing it to the other non-cached accesses in this benchmark.

For the 64-bit HW implementation, both devices are limited by the low AXI bus frequency of 110 MHz resulting in a bandwidth of 839 MiB/s. By using all four HP ports or a 256-bit F2S port, higher bandwidths of up to 3337 MiB/s for the Zynq-7020 and up to 2590 MiB/s for the Cyclone V SoC can be achieved. The difference is caused by the higher memory data rate for the Zynq-7020 of 1066 MT/s. It can also be seen that the 256-bit F2S port of the Cyclone V SoC requires a higher block width to reach its maximum bandwidth. Both devices behave similarly when using two 64-bit ports in parallel, reaching a maximum of 1644 MiB/s (Cyclone V SoC) and 1689 MiB/s (Zynq-7020), respectively. In particular, for small block widths it turns out to be more reasonable to use two 64-bit ports than using one 256-bit port.

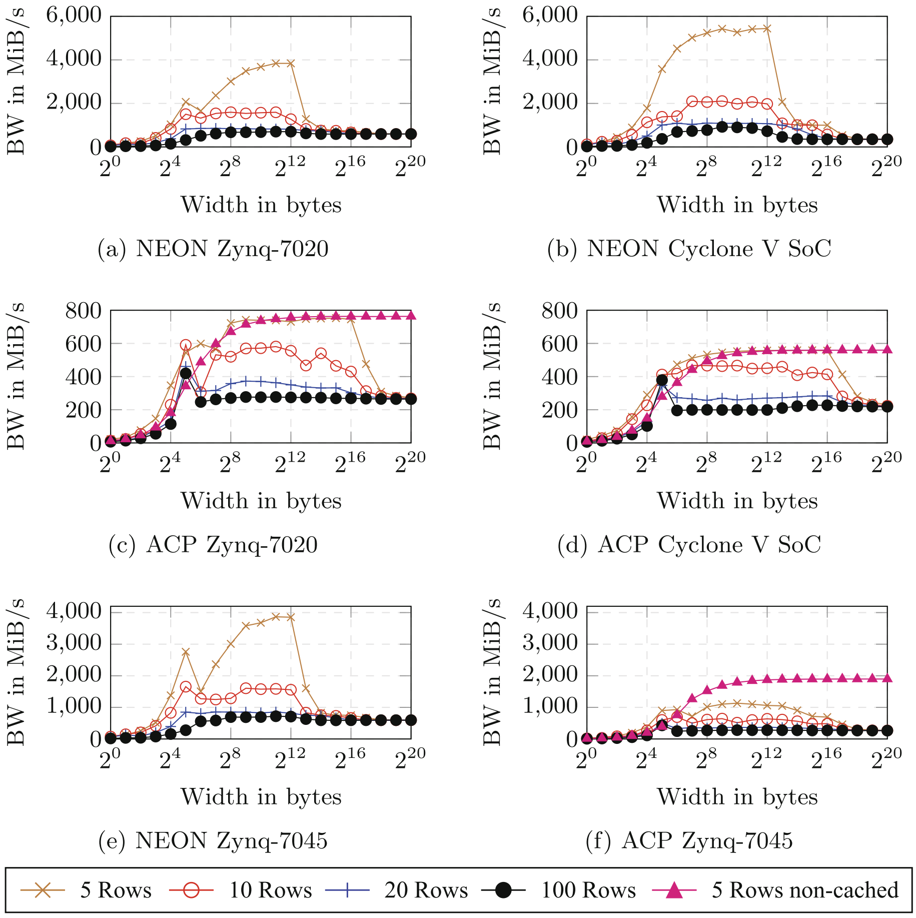
Figure 2 also shows the writing results for the same settings, again for the software and non-ACP hardware scenarios. The main difference is the improved cached SW baseline for both devices. For the Cyclone V SoC it is even comparable to the 256-bit HW implementation. In general, for the HW implementations, the same behavior as for reading can be seen: The 64-bit implementation is limited by the AXI interconnect frequency, while the 256-bit solution of Xilinx outperforms the Intel one.

The plots (e) and (f) in Fig. 2 show the result of reading and writing in parallel. As the read and write signals of an AXI interface are independent from each other, both operations can be performed simultaneously. This has been accomplished by instantiating a read and a write engine in parallel. For the 64-bit and the 2×64-bit HW implementations, the bandwidth has increased significantly. This is caused by the increase of the bus width: As two independent data busses are used for reading and writing, the effective bus width is doubled.

While the former experiments deal mostly with non-coherent accesses, Fig. 3(a)-(d) compares reading from main memory using the ACP in coherent mode running at 100 MHz to the NEON-accelerated SW baseline. The chosen scenario uses a stride of 1 MiB and a height of 5, 10, 20 or 100 rows. The different heights are required to analyze the impact of the cache architecture on the bandwidth. To see the full impact of caching, the same operation has been performed 100 times before starting the actual measurements as this reduces the number of cold cache misses.

For the SW baseline it can be seen that caching is especially useful for small heights. For a height of 5 rows and a fixed width of 4096 bytes a bandwidth of 3839 MiB/s and 5441 MiB/s can be seen, respectively. On the other hand, for larger heights some rows are removed from cache due to conflicting cache misses, which results in a higher miss rate. In fact, for small widths it is even possible on the Cyclone V SoC to achieve bandwidths higher than the maximum DDR bandwidth of 3052 MiB/s.

For the ACP, the bandwidth is significantly lower compared to the SW baseline. The data bus width of 8 bytes and the employed frequency of 100 MHz



(a) NEON Zynq-7020

(b) NEON Cyclone V SoC

(c) ACP Zynq-7020

(d) ACP Cyclone V SoC

(e) NEON Zynq-7045

(f) ACP Zynq-7045

Fig. 3. The cached read bandwidth (BW) for a fixed stride of 1 MiB. Note the different scale for the Zynq-7045. For all scenarios, the same transactions have been performed 100 times before starting the measurement in order to fill the caches and therefore maximize the throughput.

limit the bandwidth to 763 MiB/s. In fact, for widths smaller than 256 bytes, a higher bandwidth can simply be reached by performing non-coherent accesses on the ACP. Anomalously high is the ACP bandwidth for a width of 32 bytes. As this behavior occurs on both devices, it indicates a general limitation of the ACP port.

Zynq-7045 and Soft-Core Memory Controller. The previous part of the evaluation deals with two low-cost FPGA-SoCs. More powerful FPGA-SoCs are also available, however. Furthermore, HW/SW-codesign can also be realized by using soft-core SoCs. In this part, the *Zynq-7045* as an example of a high-performance FPGA-SoC as well as Xilinx's soft-core memory controller are evaluated. The same benchmarks as before have been used. The *ZC706 Evaluation Board* from Xilinx has been employed for evaluation.

The results for the Zynq-7045 are depicted in Figs. 2(g)-(i) and 3(e)-(f). While the memory ports are the same as for the Zynq-7020, a higher frequency of 214 or even 250 MHz for the engines and the AXI bus can be achieved. As a result, the bottleneck when using the four HP ports in parallel is not located in the AXI interconnect as before, but caused by the maximum bandwidth of the memory controller of 4066 MiB/s. Furthermore, when comparing the ACP benchmark results for all three FPGA-SoCs, it can be seen that the advantage of using non-coherent accesses for larger blocks compared to coherent accesses is even more significant for the Zynq-7045. Besides these aspects, the results for the Zynq-7045 qualitatively match the results for the Zynq-7020.

In order to evaluate the memory bandwidth of a HW/SW-codesign running on a soft-core SoC, a soft-core memory controller [11] has been evaluated. This memory controller can be instantiated in various Xilinx FPGAs which are connected to DDR memory. In this case, the same ZC706 board as before has been used. However, instead of using the memory connected to the hard-core memory controller of the Zynq, an external 1 GB DDR3 SODIMM is connected to the soft-core memory controller. As a result, the Zynq-7045 behaves like an ordinary FPGA in this evaluation, i.e. one without a hard-core CPU.

As the memory controller is highly configurable, it can use an AXI bus with a data width of up to 512 bits. The design could be placed and routed with a maximum frequency of 166 MHz for the AXI interconnect, resulting in a maximum read or write bandwidth of 10132 MiB/s. In fact, as the ZC706 Evaluation Board offers an SODIMM with a data rate of 1600 MT/s and a bus width of 64 bits, a maximum bandwidth of even 12207 MiB/s could be obtained in theory. The same synthetic benchmarks as for the hard-core memory controllers have been evaluated, resulting in a peak bandwidth of 9230 MiB/s for reading and 8754 MiB/s for writing. This is significantly higher than the maximum memory bandwidth for any of the current FPGA-SoCs.

5.2 H.265/HEVC Trace-Based Benchmark

HEVC motion compensation has been evaluated as a representative real benchmark. It processes blocks (i.e. parts of video frames) of size between 4×2 and 128×64 bytes. As it also requires a different number of neighboring pixels of these blocks, it actually has to read blocks of size between 7×5 and 142×71 bytes. Furthermore, it has to write blocks between 32×32 and 128×64 bytes.

A trace of the application's memory transactions has been generated. Afterwards, these memory accesses have been performed on different FPGA-SoCs. The results are depicted in Fig. 4. On the two Zynq systems, two or four HP ports have been used to process different parts of the same frame in parallel. Otherwise, each frame has been processed sequentially. For benchmarking the Zynq-7045, a frequency of 214 MHz has been employed, with a frequency of

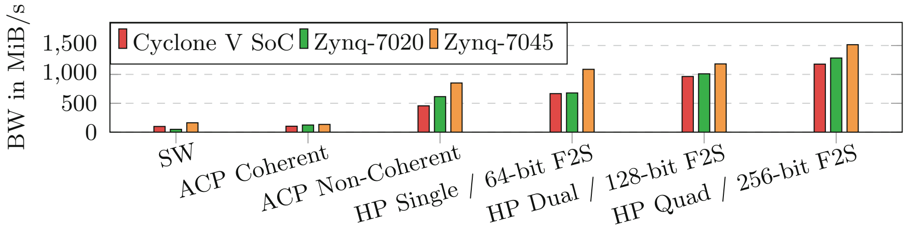


Fig. 4. The achievable read bandwidth (BW) for a trace-based simulation of the memory accesses of the motion mompensation stage of an H.265/HEVC decoder. A Full HD video stream with a medium bitrate has been used.

100 MHz for the other two SoCs. Again, these frequencies pose the maximum on each device for this implementation.

It can be seen that both the SW baseline and the coherent ACP implementation offer a very low bandwidth of less than 200 MiB/s. In comparison, non-coherent HW solutions offer a significantly higher throughput. While the bandwidth does not scale perfectly with the number of ports (Zynq) or the port width (Cyclone V), it allows to increase the bandwidth significantly this way. As the difference for 256-bits between the 100 MHz solution on the low-cost FPGA SoCs and the 214 MHz solution on the Zynq-7045 is rather small, the bottleneck is apparently not located in the AXI bus, but instead in the memory controller itself. For the HP Quad solution on the Zynq-7045, a bandwidth of 1515 MiB/s can be reached, which is sufficient for real-time Full HD decoding [12].

The theoretical maximum of 4066 MiB/s on the Zynq cannot be reached, however. This can be explained with the different block sizes: As can be seen in Fig. 2(g), the expected bandwidth when using four HP ports is below 1000 MiB/s for those blocks with the smallest width (5 bytes) in this workload. On the other hand, a bandwidth of almost 4000 MiB/s can be reached for those blocks with the largest width (142 bytes). As a result, the actual bandwidth is in between these two extremes. An analysis of the block sizes for the workload shows that almost 50% of the blocks have a width smaller than 16 bytes and more than 80% of the blocks have a width smaller than 32 bytes. Therefore, the small memory accesses dominate which results in a relatively low bandwidth.

6 Conclusions

In this paper, three different FPGA-SoCs from Xilinx and Intel have been evaluated regarding their memory bandwidth. In particular, two low-cost devices, the *Zynq-7020* from Xilinx and the *Cyclone V SoC* from Intel, have been compared. The *Zynq-7045* from Xilinx has been evaluated as an example for a high-performance FPGA-SoC. By using several synthetic benchmarks, it has been possible to determine the memory bandwidth for various scenarios. A real workload from the field of video coding has been applied as well. Finally, the bandwidth of these devices has been compared to the bandwidth of a soft-core memory controller.

The following general conclusions can be drawn:

- For bandwidth-demanding applications like H.265/HEVC motion compensation, HW/SW-codesigns on recent FPGA-SoCs have the potential to significantly outperform SW solutions running on the same CPU.
- High-performance FPGA-SoCs like the *Zynq-7045* offer significantly higher bandwidth than low-cost devices. However, the maximum bandwidth of the memory controller of 4066 MiB/s can pose a bottleneck in this case.
- For applications with demanding memory bandwidth requirements and moderate CPU performance requirements, a soft-core SoC system might be a reasonable choice as it offers up to 9230 MiB/s.

References

1. Fu, H., Clapp, R.: Eliminating the memory bottleneck: an FPGA-based solution for 3D reverse time migration. In: 19th ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA), Monterey, USA (2011)
2. Naylor, M., Fox, P., Markettos, A., Moore, S.: Managing the FPGA memory wall: custom computing or vector processing? In: 23rd International Conference on Field Programmable Logic and Applications (FPL), Porto, Portugal (2013)
3. Dobai, R., Sekanina, L.: Image filter evolution on the Xilinx Zynq platform. In: NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Turin, Italy (2013)
4. Ishikawa, S., Tanaka, A., Miyazaki, T.: Hardware accelerator for BLAST. In: 6th IEEE International Symposium on Embedded Multicore SoCs (MCSoC), Aizu-Wakamatsu, Japan (2012)
5. Sadri, M., Weis, C., Wehn, N., Benini, L.: Energy and performance exploration of accelerator coherency port Using Xilinx Zynq. In: ACM 10th FPGAWorld Conference, Copenhagen, Denmark, Stockholm, Sweden (2013)
6. Sklyarov, V., Skliarova, I., Silva, J., Sudnitson, A.: Analysis and comparison of attainable hardware acceleration in all programmable systems-on-chip. In: 2015 Euromicro Conference on Digital System Design (DSD), Funchal, Portugal (2015)
7. Tahghighi, M., Sinha, S., Zhang, W.: Analytical delay model for CPU-FPGA data paths in programmable system-on-chip FPGA. In: 12th International Symposium on Applied Reconfigurable Computing (ARC), Mangaratiba, Brazil (2016)
8. Zynq-7000 All Programmable SoC Technical Reference Manual by Xilinx. http://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf
9. Altera's User-Customizable ARM-based SoCs by Altera. http://www.altera.com/literature/br/br-soc-fpga.pdf
10. Sullivan, G., Ohm, J.-R., Han, W.-J., Wiegand, T.: Overview of the High Efficiency Video Coding (HEVC) standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1649–1668 (2012)
11. 7 Series FPGAs Memory Interface Solutions User Guide by Xilinx
12. Chi, C.C., Alvarez-Mesa, M., Bross, B., Juurlink, B., Schierl, T.: SIMD acceleration for HEVC decoding. IEEE Trans. Circuits Syst. Video Technol. **25**, 841–855 (2014)