Memory prefetching for tomography acceleration on FPGAs through HLS tools

Daouda Diakite, Nicolas Gac

Memory prefetching for tomography acceleration on FPGAs through HLS tools

Daouda Diakite, and Nicolas Gac
Université Paris-Saclay, CNRS, CentraleSupélec, L2S, 91190, Gif-sur-Yvette, France.

Abstract

Backward projection is one of the most time-consuming steps in method-based iterative reconstruction computed tomography. The 3D back-projection memory access pattern is potentially enough to efficiently exploit the computation power of acceleration boards based on GPU or FPGA. However, exploiting the full potential of these architectures has always been a major concern. Therefore, an algorithm architecture co-design approach is necessary to harness these parallel architectures sufficiently. This paper proposes an OpenCL acceleration of the voxel-driven 3D back-projection algorithm on an Arria 10 FPGA. We perform an offline study of the algorithm memory access pattern to prefetch sinogram data to the on-chip BRAM before performing reconstruction.

1 Introduction

X-ray computed tomography(CT) is an imaging technique that initially found its application in the medical field. It has been extended to industrial applications such as non-invasive human body investigation and non-destructive testing of industrial materials. Model-Based Iterative Reconstruction (MBIR) algorithms are proved to produce better image quality at the cost of expensive computational time. To reduce the reconstruction time of CT algorithms, hardware accelerators are required.

FPGAs based on HLS tools are experiencing great consideration as an acceleration platform for many applications such as high-performance computing [1, 2, 3]. The maturity of their architectures and many built-in floating-point units (DSPs) in the latest FPGAs explain this interest. These floating-point units provide high design flexibility and are optimized to support high-performance DSP applications in IEEE 754 compliant floating-point single precision.

2 3D back-projection algorithm

3D Computed Tomography (CT) aims to acquire the internal density of 3D objects from external measurements called sinogram. An object (3D volume) is placed between an X-ray source and a detector plane as illustrated Fig. 1.

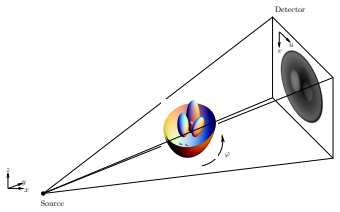

Figure 1: X-RAY CT Projection

This work is focused on the 3D back-projector operator used in tomography reconstruction.

3 Offline Memory Access Analysis

The projection data (sinogram) size is tremendous and cannot fit in FPGA on-chip memory. The block of voxels reconstruction will be wise to avoid global memory bottleneck and achieve better performance. The projection block of voxels (B_x, B_y, B_z) corresponds to a rectangle shape $(local_u, local_v)$ in the detector plane (Fig. 2) for a given projection angle φ_i. A high data re-utilization exists and is even more important for neighboring voxels. For each projection angle, voxels in the same block will access the same sinogram tile. The main concern is to capture the sinogram footprint without losing information and calculate the coordinates of its endpoints. For each voxel (x, y, z), the reconstruction depends on its N_φ angular projections, these projections are spatially distant due to their storage in the sinogram following the order (u, v, φ). To increase the spatial locality, reconstruction by a group of voxels in the same block is beneficial. Instead of looking for the contribution of a single voxel in the sinogram data for a given projection angle, one could look for the contribution of all the voxels in the block, which will always be in the same memory zone thanks to the CT system geometry. This makes access to memory contiguous and regular to take advantage of BRAM memory by prefetching the sinogram data. The new implementation uses

the BRAMs memory extensively with access latency reduced, and automatic caching is avoided.

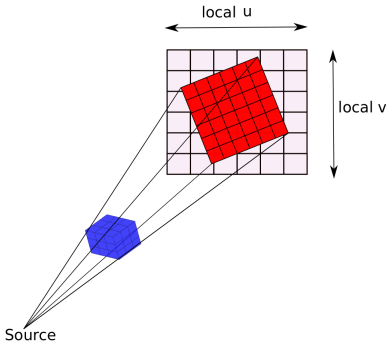

Figure 2: Block of voxels reconstruction

Version		Stall (%)	Occ (%)	Freq (Mhz)	Time (s)
BP-Cache		71.27	24.6	150	3.65
BP-Prefetch	$64^2 \times 1$	2.79	43.5	183	0.843
	$64^2 \times 2$	0.34	58.9	186	0.658
	$64^2 \times 4$	0.2	74.4	187	0.529
	$64^2 \times 8$	0.06	84.1	189	0.425

Table 1: Block size variation effect on the performance

	FF	LUT	BRAM	DSP
BP-Cache	452579(29%)	188411 (24%)	1758(65%)	406(27%)
BP-Prefetch	407183(25%)	184616(23%)	1952(62%)	949(63%)
Available	1577720	788860	2537	1518

Table 2: FPGA resources consumption on Arria 10

4 Results and discussion

In our implementation, the critical path consists of reconstructing the block of voxels with the innermost loop over the voxels. The loop body, considered as processing element (PE), can be replicated for parallel voxel intensity computation by loop unrolling. It is then possible to have 64 PEs, with stall-free access to local memory, in our architecture without exceeding available resources. We used for this experiment the Intel Arria 10 GX FPGA (10AX115N2F45E1SG) with 1150K logic elements.

Table 1 shows the results of our implementation, the BP-cache design is an optimized version of the 3D back-projection using burst-coalesced cached LSU from [4] running on Arria 10. OpenCL optimizations such as loop pipelining, loop unrolling were applied to this version to accelerate the reconstruction time. This version suffers from a high pipeline stall percentage because of the memory access pattern making the global bandwidth the main bottleneck. The implementations based on blocks of voxels using BRAM memory achieve better performance compared to the BP-cache version. In the BP-Prefetch design, all loops are pipelined with an Initiation Interval (II) of 1. Once B_x and B_y are fixed, B_z tuning allows to optimize the data reuse rate and therefore the overall performances as illustrated on table 1.

We have obtained a better execution time with the $64 \times 64 \times 8$ block, which corroborates the static study performed on the data reuse. The pipeline occupancy is 84.1% with a 0.06 stall percentage, which characterizes an excellent utilization of the FPGA device. Even when designing in HDL language, the purpose is to implement a pipeline with a 0% stall percentage. Compared to the previous BP-cache design, we achieved a speedup of 8.6 at 188.9MHz.

The resource usage of the $64 \times 64 \times 8$ block version is presented in table 2. As mentioned above our design contains 64 PEs, the resource consumption is compared to the BP-Cache design.

5 Conclusion and future work

We present in this paper an optimization based on the back-projection algorithm for CT reconstruction using FPGA BRAM efficiently. A reconstruction by blocks of voxels was developed to maximize data reuse and reduce external memory bandwidth and maximize at the same time the use of the on-chip local memory. We plan to run the design on Intel Stratix 10 to express more parallelism for future work and compare it to other HLS tool implementations.

References

[1] F. B. Muslim *et al.*, "Efficient FPGA implementation of OpenCL high-performance computing applications via hls," in *IEEE Access*, vol. 5, 2017.

[2] H. R. Zohouri *et al.*, "Evaluating and optimizing OpenCL kernels for high performance computing with FPGAs," in *SC '16: Proceedings of the International Conference for HPC, Networking, Storage and Analysis*, 2016.

[3] M. A. Mansoori *et al.*, "Efficient FPGA implementation of PCA algorithm for large data using hls," in *PRIME Conference*, 2019.

[4] M. Martelli *et al.*, "3D Tomography back-projection parallelization on Intel FPGAs using OpenCL," *Journal of Signal Processing Systems*, 2018.