

(Extra quiz credit for everyone) Find a published paper from an ACM or IEEE conference that discusses a novel sparse matrix format that was not covered in class. Discuss why the proposed format is superior to the CSR or CSC format. Make sure cite your sources.

Modified Compressed Sparse Row Format for Accelerated FPGA-Based Sparse Matrix Multiplication

(Cited from <https://ieeexplore.ieee.org/document/9181266>)

This paper presents a modified CSR sparse matrix format, which improve SMVM performance on multi-core FPGA-based hardware accelerators environment. As far as I know, FPGA (Field Programmable Gate Array) is a further development of programmable devices such as PAL (Programmable Array Logic) and GAL (General Array Logic). It emerged as a semi-custom circuit in the field of ASICs, addressing the shortcomings of custom circuits and overcoming the limited number of gates of the original programmable devices. As a consequence, FPGAs' pipelinability combined with parallelism become more competitive in SMVM implementation.

As we all know, large size sparse matrixs are essential for SMVM algorithm. CSC or CSR format store sparse matrices by column and by row respectively and basically they're always used to reduced memory and enhance efficiency of data communication between cores and memory. However, these two traditional storage methods may increase the computing load per core considerably when the volume of tasks increases significantly. Correspondingly, MSCR is to make optimizaitons on the memory bandwidth and thus increase performance through eliminating the bottleneck.

The advantage of this format over CSR and CSC is that it is based on the Hu's scheduling CSR format (T. C. Hu, "Parallel sequencing and assembly line problems" Operations research, vol. 9, no. 6, pp. 841–848, 1961.) and accelerates the most time-consuming computational tasks to the maximum extent possible. The algorithmic procedure is to reallocate and store the non-zero values of a particular sparse matrix in memory over a certain number of CPU cores, so that each row can be processed by a separate core. This paper uses a System Verilog model to evaluate the performance. Experimental text files are loaded to the MAC in ModelSim, and compared in terms of density, size and acceleration. Due to the advantage of the parallelizaation capability of FPGAs, the paper's method has a significant speed-up compared to traditional storage methods.