# A Model for Peak Matrix Performance on FPGAs

Colin Y. Lin and Hayden K.-H. So
*Dept. of Electrical & Electronic Engineering*
*The University of Hong Kong*
*Hong Kong*
*Email: {linyu, hso}@eee.hku.hk*

Philip H.W. Leong
*School of Electrical and Information Engineering*
*The University of Sydney*
*Sydney, Australia*
*Email: philip.leong@sydney.edu.au*

***Abstract*—Computations involving matrices form the kernel of a large spectrum of computationally demanding applications for which FPGAs have actively been utilized as accelerators. The performances of such matrix operations on FPGAs are related to underlying architectural parameters such as computational resources, memory and I/O bandwidth. A model that gives bounds on the peak performance of matrix-vector and matrix-matrix multiplication operations on FPGAs based on these parameters is presented. The architecture and efficiency of existing implementations are compared against the model. Future trends in matrix performance on FPGA devices are estimated based on the performance model and system parameters from the past decade.**

## I. Matrix Operations on FPGAs

For the purpose of this work, we assume that initially all input data are stored off-chip and must be loaded onto the FPGA fabric during the operation. Similarly, the result of computation must be stored off-chip. We make no assumption about the mechanism of data I/O as that is irrelevant to the derivation of the performance model. Furthermore, we assume the input matrices and vectors are large.

The following architectural parameters were considered in this work: on-chip logic/DSP resources ($r$), on-chip memory ($m$), I/O bandwidth ($b$), number of MACCs ($k$), and clock frequency ($f$).

In a matrix-vector multiplication, $y = Ax$, the entire input vector $x$ must be reused in the calculation of each of the element in $y$ while each row of data in $A$ is used only once. Therefore, $x$ is stored using on-chip memory while $A$ is streamed through the FPGA to generate results. Blocking is used if the on-chip memory is limited.

The performance bound is found to be determined by the values $kf$ and $b$. If $kf < b$, the performance is computation resource bounded and the maximum achievable operation per second equals to $\mathbf{OPS}_{max} = \left\lfloor \frac{R}{r_{macc}} \right\rfloor \cdot 2f$. If $b < kf$, then the performance is bounded by I/O bandwidth. In this case, $\mathbf{OPS}_{max} = 2B$.

In a matrix-matrix multiplication, $C = AB$, every data element from both matrix $A$ and $B$ are reused more than once. Therefore, the amount of on-chip memory $m$ that may be used as temporary storage is an important design parameter. Optimized I/O scheme must therefore be used to utilize on-chip memory efficiently for the input matrices $A$ and $B$.

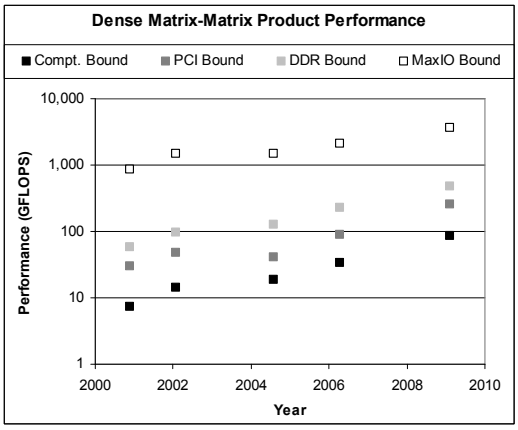


Figure 1. Matrix-matrix multiplication performance bounds for Xilinx Virtex series FPGAs.

In this work, the on-chip memory is mainly used to store the partial results of matrix $C$. The values are updated in-place as $A$ and $B$ are streamed on and off chip for computation.

With such assumptions, the performance of matrix-matrix multiplication have been found to depend on $k$, $f$, $m$ and $b$. If $2kf < \sqrt{m}b$, the performance is bounded by compute resource availability. In this case, the maximum performance is $\mathbf{OPS}_{max} = \left\lfloor \frac{R}{r_{macc}} \right\rfloor \cdot 2f$. If $\sqrt{m}b < 2kf$, then the performance is bounded by on-chip memory and I/O bandwidth. In this case, $\mathbf{OPS}_{max} = \sqrt{M}B$.

## II. Technology Trends & Conclusion

Figure 1 shows the memory, I/O and computation bound on matrix-matrix multiplication for Xilinx Virtex series FPGAs over the past decade. Computational resources have been the limiting factor in the past. However, with the amount of logic resources increasing according to Moore's Law, it is possible that I/O and on-chip memory will become the dominant bound in the near future.

978-0-7695-4301-7/11 $26.00 © 2011 IEEE
DOI 10.1109/FCCM.2011.51