Correlation Analysis of Functional Magnetic Resonance Imaging Data using FPGA hardware

M. Gipp[1], D. Gembris[1], G. Lienhart[2], A. Kugel[2], R. Männer[1,2]

[1]Institut für Computerunterstützte Medizin, Universität Mannheim, Mannheim, Germany, [2]Lehrstuhl für Informatik V, Universität Mannheim, Mannheim, Germany

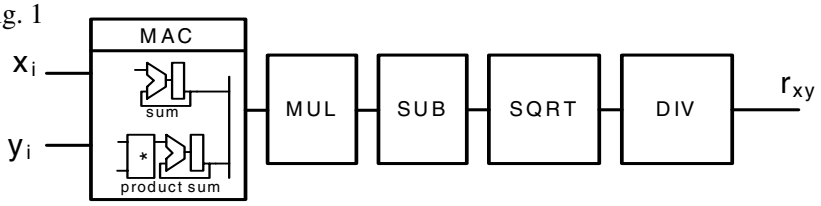
Introduction: Functional magnetic resonance imaging (fMRI) generates series of 3D data sets, representing the measured BOLD signal time courses of each voxel. Correlation analysis compares the time-course of voxels or ROIs (region of interest) to a stimulus function or to other signal time courses, in case of functional connectivity analyses [1]. When high resolution correlation maps are desired this can imply to correlate the time-series of each voxel with all others, resulting in high computational demands ($O(N^2)$ correlations for N time series). To reduce the resulting computation time we propose the use of field programmable gate arrays (FPGA), reconfigurable logic, which can be tailored for given computational problems. A speed-up of the computation would open new perspectives for real-time analysis or interactive post-processing of functional connectivity data. The work relies on hardware based on XILINX FPGAs previously developed at our institute, that has already been used for scientific computations [2].

Methods: For the hardware implementation, a standard PC is used equipped with our PCI-plugin card, called MPRACE. In addition to the hardware design a software program is currently being developed that sends the fMRI data to the MPRACE board, with 512MB (extendable to 1GB) on-board memory, that controls the computation and that visualizes the resulting maps. The following equation forms the basis of our FPGA design (currently neglecting varying relative delays between fMRI time series):

(1)
$$r_{xy} = \frac{\sum_{i=1}^{T} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\left[\sum_{i=1}^{T} (x_i - \bar{x})^2\right]\left[\sum_{i=1}^{T} (y_i - \bar{y})^2\right]}}$$

(2)
$$r_{xy} = \frac{T\sum_{i=1}^{T} x_i y_i - \sum_{i=1}^{T} x_i \sum_{i=1}^{T} y_i}{\sqrt{\left[T\sum_{i=1}^{T} x_i^2 - \left(\sum_{i=1}^{T} x_i\right)^2\right]\left[T\sum_{i=1}^{T} y_i^2 - \left(\sum_{i=1}^{T} y_i\right)^2\right]}}$$

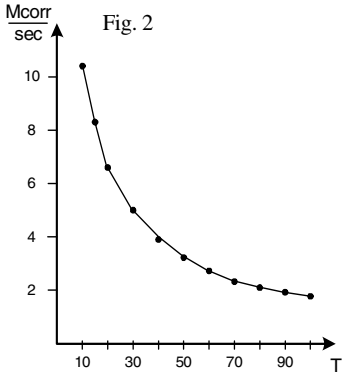
Fig. 1



The first expression (Eq. 1) shows the common correlation formular, where x_i denotes the reference time-course and y_i a given voxel time series. Eq. (2), where the averages have been substituted by sums, is equivalent to Eq. (1), but allows a more efficient hardware implementation, since less chip-resource expensive division operations are needed. T is the number of sampling points per time-course and has a typical range of 10 to 100. Figure 1 shows a schematic diagram of the hardware design comprising five blocks, over which the evaluation of Eq.2 is distributed. The MAC block is responsible for calculating all the sums including product sums. In the consequent MUL block products of the sums with each other and the constant T are calculated and the subtraction operations follow in the SUB block. The SQRT block calculates the denominator, and the DIV block the resulting correlation value. The MAC block works serially, requiring multiple clock cycles. All other blocks in contrast work in parallel, generating one result per cycle. To reduce the amount of data generated by the correlation ($O(N^2)$), the correlation results will be averaged to obtain a mean correlation map (only sum without division by constant T).

Results: In an optimized software reference implementation, the calculation speed was measured as 10.5 million correlations per second (MCorr/s) (3GHz intel Pentium 4, 1GB RAM) for T = 10. The computation time depends on T: larger T results in slower speed as shown in (4). Our hardware design allows to calculate one correlation per cycle if the MAC block is replicated ten times, for T = 10. This results in the maximum speed of 100 MCorr/s at 100 MHz clock rate, which is typical for FPGA chips. Depending on the available logic resources, duplicating the circuit sketched in Fig.1 can increase the speed. Next generation Virtex4 FPGA chips have between two and six times more resources than Virtex2 chips and they also support higher clock frequencies, which will make it possible to accelerate the computations accordingly.

Fig. 2



Discussion: The hardware-supported computation will be at least one order of magnitude faster than a pure software implementation. The increased speed makes our development interesting as an explorative tool for functional connectivity. So far it relies mainly on hypothesis testing approaches, which principally bares the risk of overlooking unexpected relationships present in the data.

References:

1. Bharat Biswal, F.Zerrin Yetkin, Victor M. Haughton, James S. Hyde. Funtional Connectivity in the Motor Cortex of Resting Human Brain Using Echo-Planar MRI, Medical College of Wisconsin, Milwaukee, Wisconsin,1995
2. Gerhard Lienhart, Andreas Kugel, and Reinhard Männer. Using Floating Point Arithmetic on FPGAs for Accelerating Scientific N-Body Simulations. In *Proc. FCCM'02, Symposium on Field-Programmable Custom Computing Machines*, page 182-191, Napa Valley, California, USA, April 2002