

CUSTOM FPGA LOGIC ARCHITECTURE FOR DSP TRANSFORMS

TEAM MEMBERS:

BHARATHWAJ V. SANKARA
sankara@wisc.edu

SHUBHIKA TANEJA
taneja@wisc.edu

PROJECT HIGHLIGHT:

We will propose a custom logic architecture targeted to speed up DSP transforms. The routing will be excluded in this. The mapping from the software to hardware for the transforms will be shown onto the custom logic blocks of the FPGA (How The Different Transforms Are Mapped to the Architecture). If time permits we will try to compare the implementation of transforms on our architecture with that on a Xilinx FPGA family.

MOTIVATIONS:

We plan to target the DSP transforms like DFT, DCT, DST, DHT and SAD computations. Each of these is used in many DSP applications. SAD computations are used in motion estimation (MPEG), 2-D DCT is used in JPEG standard. DFT is used in frequency analysis and studying the frequency spectrum of signals. All these transforms are the base for many DSP algorithms and they are frequently used.

The FPGAs provide the reconfigurability advantage over DSPs. The same hardware can be reconfigured for implementing different applications. The use of FPGA provides high degree of parallelism and flexibility.

The FPGA architecture that we propose will be designed to speed up the transforms mentioned. We will show how the four transforms and SAD can be mapped to the same hardware. We expect our custom architecture to perform better than the standard FPGAs available. Some of the standard FPGAs have DSP slices, adders and multipliers. These help speed DSP applications. Our architecture specifically targets the transforms so it should outperform the available FPGAs for these transforms implementation.

APPROACH:

1. TRANSFORMS TO LOOK FOR: *Transform functions* are very frequently used in signal processing. Different Applications use different transforms, like JPEG uses DCT, DHT used in Image processing and DFT used in Spectral Analysis of signals. The basic operation performed in all of these transforms is the MAC operation. We will look at different transforms that can be computed using similar functional units.

2. DFG OF THE LOGIC BLOCK FOR OUR custom-FPGA: After deciding the transforms, we plan to derive a '*Data Flow Graph*' for functional unit, which is going to be the logic block of the custom-FPGA.

3. OPTIMIZING THE DFG: Then we plan to optimize the DFG using different optimizing techniques like unfolding, pipelining and re-timing. We also plan to look at some systolic array architecture to implement the logic blocks.

4. RTL OF THE LOGIC BLOCK: After optimizing the DFG we plan to design the RTL of the logic block using some HDL and synthesize them to obtain the area constraint. Serial bit and parallel architecture style will be decided and techniques like unfolding might be employed for this.

5. MAPPING SOFTWARE ALGORITHM TO HARDWARE: We propose a method to map the software to the custom-FPGA using some compiler optimization techniques. These might involve using some loop transformation techniques.

6. PERFORMANCE COMPARISON: Finally if time permits we would compare the performance of our custom FPGA with a standard FPGA implementation.

PRIOR WORK:

The following papers are relevant to our project:

1. Katherine Compton and S. Hauck: "Reconfigurable Computing: A Survey of Systems and Software", IEEE Transactions on Volume 10, Issue 3, June 2002.
2. E.P. Mariatos, M.K. Birbas and A.N. Birbas: "A Reconfigurable DSP Board Based On CORDIC Elements", Proceedings of IEEE Workshop on Rapid System Prototyping, July 1993.
3. Chris Dick: "Computing The Discrete Fourier Transform on FPGA Based Systolic Arrays", Proceedings Of The Fourth International ACM Symposium on Field Programmable Gate Arrays, FPGA'96.