

January 01, 2007

Phase unwrapping using reconfigurable hardware

Sherman Braganza
Northeastern University

Miriam Leeser
Northeastern University

W. C. Warger II
Northeastern University

C. M. Warner
Northeastern University

C. A. DiMarzio
Northeastern University

Recommended Citation

Braganza, Sherman; Leeser, Miriam; Warger II, W. C.; Warner, C. M.; and DiMarzio, C. A., "Phase unwrapping using reconfigurable hardware" (2007). *Research Thrust R3 Presentations*. Paper 14. <http://hdl.handle.net/2047/d10009085>

This work is available open access, hosted by Northeastern University.



Phase unwrapping using Reconfigurable Hardware

Sherman Braganza, Miriam Leeser, W.C. Warger II, C.M. Warner, C. A. DiMarzio

sbraganz@ece.neu.edu

mel@ece.neu.edu



This work was supported in part by Gordon-CenSSIS, the Bernard M. Gordon Center for Subsurface Sensing and Imaging Systems, under the Engineering Research Centers Program of the National Science Foundation (Award Number EEC-9986821).

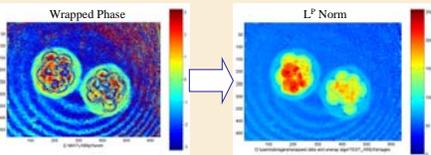
Goal

Accelerate the performance of the minimum L^p Norm phase unwrapping algorithm using Field Programmable Gate Arrays (FPGAs)

Abstract

The most computationally intensive part of the minimum L^p Norm phase unwrapping algorithm[1] (its kernel) is the 2D Discrete Cosine Transform (DCT) that computes the variable p in the equation $Qp=c$ using the Pre-conditioned Conjugate Gradient (PCG) method. The separability of the DCT means that the 2D transform can be decomposed into a series of 1D DCTs that compute the transforms of the rows followed by the transforms of the columns. Furthermore, the DCT can be expressed in terms of a Fast Fourier Transform (FFT), which allows the hardware implementation to use a pre-designed FFT core. This poster presents a design that implements the 1D DCT on a Xilinx FPGA that is part of the Wildstar II Pro board.

This implementation performs 1D DCTs on large block sizes using a block floating point format. The DCT was designed to use fewer resources than other popular approaches due to the larger point sizes supported which would otherwise consume all available chip area, but at the cost of higher latency. This latency is similar to that required for an identically sized FFT. A 512-point DCT has been shown to take 1771 cycles or 13.3 us at 133 MHz as compared to a similarly sized FFT that takes 1757 cycles or 13.2 us (including full data load and unload times).



Algorithm

The procedures for implementing a DCT vary depending on the direction of the transform. For the forward transform:

- 1) Form a shuffled sequence v from the input x .
- 2) Take the DFT of v to get V .
- 3) Multiply $V(k)$ by $2 \exp(-j\pi k/2N)$. The real part forms $X(k)$ and the negative of the imaginary forms $X(N-k)$.

Similarly for the inverse transform:

- 1) Build $V(k)$ from $X(k)$ and multiply by $2 \exp(j\pi k/2N)$
- 2) Compute the IDFT of V
- 3) Perform the inverse of the original shuffle to get x .

Components

SHUFFLE

- Latency of one cycle
- Ordering is performed by 'mirroring' even and odd inputs around $N/2$ as shown below in Fig. 1.
- Inverse of shuffle is performed in the inverse DCT

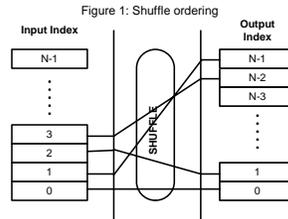
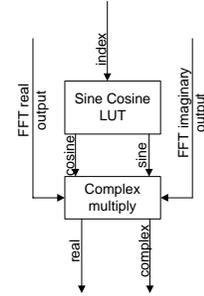


Figure 1: Shuffle ordering

FFT

- 24 bit, block floating point
- BlockRAM storage for twiddle factor storage
- Sub 13 us minimum latency for 512 point transform. 1757 cycle latency.
- Radix-4, supports both forward and inverse transforms.
- Run-time configurable transform length

Figure 2: Rebuild Rotate data flow



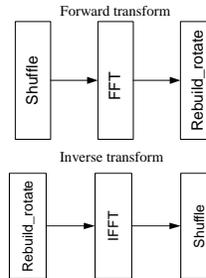
REBUILD ROTATE

- Using decomposition of $\exp(-j\pi k/2N)$ into sines and cosines
- Sine Cosine calculation is done via LUTs

Dataflow

Based on the above breakdown, the design was decomposed into the components displayed to the right. The dataflow is indicated by the arrows and is handled by a controller (not shown).

The two operations are closely related allowing for the reuse of much of the functionality. This reduces design area at the cost of added complexity for the controller.



Results

- Achievable clock speeds of 140 to 150 MHz on a V2P70-6 for sizes ranging from $N=32$ to $N=1024$
- Small size relative to other implementations in the field. About 30% of chip area for a 1024 point transform
- Higher latency due to serialized nature of the algorithm
- 14 BlockRAMs and 48 multipliers used out of 328 total
- 14 cycle fixed overhead above that of the core FFT
- Area requirements for all components scale with transform size except for the complex multiply since bitwidth is constant

Reconfigurable Hardware

In order to fully exploit the parallelism exposed in the algorithm, a sufficiently large FPGA with multiple banks of off-chip SRAM is needed. The Wildstar II Pro was selected as the reconfigurable solution that fulfills all these requirements.

Features of the WILDSTAR™-II PRO/PCI boards:

- Uses two Xilinx® Virtex-II Pro™ FPGAs XC2V70 (33088 slices and 5904Kb BlockRAM)
- 12 ports of DDR II SRAM totally 48MBytes, 2 ports of DDR SDRAM totally 256 MBytes
- 11 GBytes/sec memory bandwidth



WILDSTAR™-II PRO/PCI



Conclusions and Future Work

The implementation of the algorithm originally presented by Makhoul [3] was implemented on an Annapolis Wildstar II Pro and the results are discussed and found to have various attractive properties for larger transform sizes. These are:

- Fixed overhead compared to an FFT
- Use of FFT core: Design performance is based on it. Large variety available with different tradeoffs.
- Small area requirements: Larger transforms possible

The next step is to apply the 1D transform to compute a 2D DCT on image data and to gauge the performance and quality of the results.

State of the art

Although no current implementation of a phase-unwrapping algorithm exists on reconfigurable hardware, much work has been done in [1][2] to develop optimal algorithms for single-processor machines.

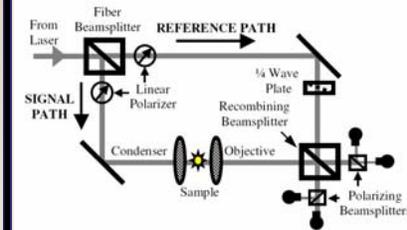
[1] D. C. Ghiglia and M. D. Pritt, *Two-Dimensional Phase Unwrapping*, John Wiley & Sons, 1998.

[2] C.W. Chen and H.A. Zebker, *Network approaches to two-dimensional phase unwrapping*, Journal of the Optical Society of America, Vol. 17 Issue 3, pp. 401-414, 2000.

[3] J. Makhoul, *A fast Cosine transform in one and two dimensions*, Acoustics, Speech and Signal Processing, Volume 28, Issue 1, pp. 27-34, 1980

Optical Quadrature Microscopy

Optical quadrature microscopy[3] was developed in 1997 based on techniques developed for coherent laser radar. A single coherent laser beam is split into two paths, one a reference and the other a signal path that passes through the sample under examination. Interference patterns are then captured by CCD cameras. Although the images could be taken with only two cameras, four are used to completely capture the entire signal including the conjugate intensities.



After the interference fringe pattern is taken, four further steps must be undertaken to produce the final image:

- 1) Phase unwrapping: Assigns integer multiples to the phase values
- 2) Term elimination: Mathematical removal of setup irregularities
- 3) Phase evaluation: Produces a phase-map from the spatial distribution of the phase
- 4) Rescaling: Converts phase to another criteria such as distance

OQM Capable Microscopes [3]

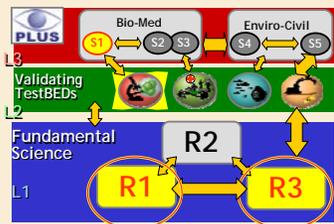
Staring Mode Microscope

- Differential Interference Contrast
- Epi-Fluorescence
- Optical Quadrature Microscopy



Keck 3D Fusion Microscope

- Differential Interference Contrast
- Epi-Fluorescence
- Optical Quadrature Microscopy
- Confocal Reflectance & Fluorescence
- Two-Photon



This work is a part of the CenSSIS BioBed effort under R1, R3 and S1. Calculating the minimum L^p norm phase unwrap takes several minutes per frame to process. This prevents the streaming of live video data from the microscope. Our goal is to develop a hardware/software implementation of phase unwrapping to achieve near real-time performance.