# Phase Unwrapping on Reconfigurable Hardware

Sherman Braganza , Miriam Leeser
sbraganz@ece.neu.edu mel@ece.neu.edu

This work was supported in part by CenSSIS, the 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

Several applications make use of coherent signals for imaging purposes:

- Synthetic Aperture Radar (SAR)
- Magnetic Resonance Imaging (MRI)
- Optical interferometry
- Adaptive beamforming, and others

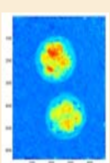
Such applications often have a reference signal to which the received signal is compared (a stable local oscillator located in the radar unit in the case of SAR) and from that comparison the phase is extracted. However, this extraction is limited by the fact that the output phase will lie between π and – π.

**Basic unwrapping:** Retrieve original phase by accumulating the differences and an integer multiple of 2π every time a discontinuity is detected. This fails in the presence of noise.

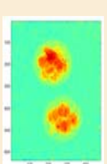
We present an initial Field Programmable Gate Array (FPGA) implementation of the core computation of the **minimum $L^P$ norm** phase unwrapping algorithm[1]. This computation involves a configurable Discrete Cosine Transform (DCT) of 512/1024 points and represents among the largest DCT/iDCTs implemented on an FPGA documented in the literature.


## Bit-width selection

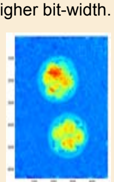
An analysis of fixed point bit-widths was performed and results compared to the double precision result on the right. Both 28 and 29 bit cases provided good approximations. The final bit-width chosen was 24 bits with a block exponent and dynamic scaling that simulates a higher bit-width.



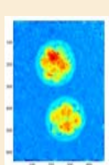
Full Precision



27 bit



28 bit



29 bit

## 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(-jπk/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(jπk/2N)*
2) Compute the IDFT of *V*
3) Perform the inverse of the original shuffle to get x.

## Data Format

- Software implementation uses floating point. Need similar accuracy, in H/W.
- Input is floating point. This is converted to fixed point + exponent for a block of data. Exponent is scaled to maximize use of dynamic range.
- After FFT, output is re-scaled using FFT block exponent and initial scaling value producing floating-point results.

## Phase unwrapping

Figure 1 and Figure 2 demonstrate the difference between a raw, wrapped embryo image and an unwrapped one. This sample originally had over eight thousand residues.

Note the phase 'jumps' in the wrapped image and the smooth transitions in the unwrapped image.

Figure 1: Wrapped Phase

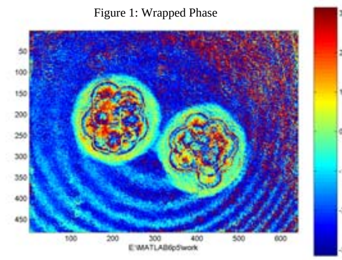
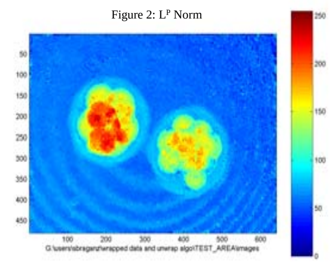


Figure 2: $L^P$ Norm



## Components

**SHUFFLE**

- Latency of one cycle.
- Ordering is performed by 'mirroring' even and odd inputs around N/2 as shown below in Fig. 3.
- Inverse of shuffle is performed in the inverse DCT.

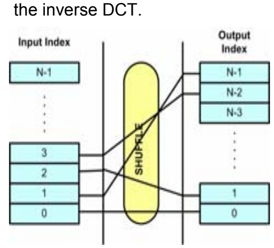



Fig. 3: Shuffle

**FFT**

- 24 bit, block floating point, run-time configurable transform length.
- 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.

**REBUILD ROTATE**

- Using decomposition of *exp(-jπk/2N)* into sines and cosines.
- Sine Cosine calculation is done via LUTs.

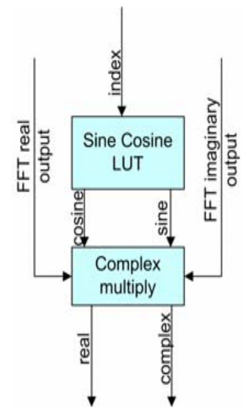



Fig. 4: Rebuild_rotate

- Complex multiplication handles rotation of input. Dataflow shown in Fig. 4.
- Total latency of 10 cycles.

**MAX TRACKER**

- Tracks and saves the maximum exponent of input data for future dynamic scaling. Single cycle latency.

**SCALE**

- Scales floating point data by MAX and converts to fixed point. Six cycle latency.

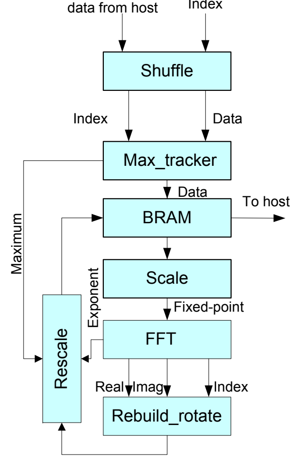
**RESCALE**

- Rescales data to original range plus the block exponent produced by the FFT. Six cycle latency.

## Dataflow

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

The forward and inverse transforms 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.




## 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 (i.e the addition of shuffle and rebuild modules).

•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.

## Results

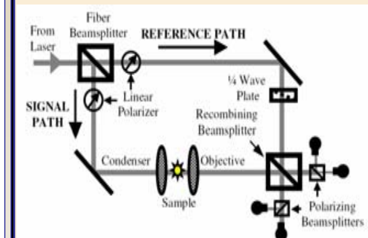
- Small size relative to other implementations in the field. About 32% of chip area for a 1024 point transform.
- 14 BlockRAMs and 48 multipliers used out of 328 total.
- Area requirements for all components scale with transform size except for the complex multiply (constant).
- Achievable clock speed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


**REFERENCES**

[1] D. C. Ghiglia and M. D. Pritt, *Two-Dimensional Phase Unwrapping*, John Wiley & Sons, 1998.
[2] W.C. Warger, *Masters thesis*, Northeastern University, 2005.
[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[2] 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.



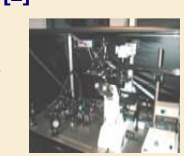

After the interference fringe pattern is taken, four further steps must be undertaken to produce the final image:

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

### OQM Capable Microscopes [2]

**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.