FPGA, ECDC Techkriti'14

Problem Statement Discussion

Problem Statement

The challenge in FPGA is to design and efficiently implement the Hilbert transformation of any given function. Hilbert Function finds its numerous applications in the field of mathematics and signal processing. It maps a time domain function to another time domain function by a convolution of the input signal with the function H(t) whose representation in frequency domain is:

$$\sigma_H(\omega) = \begin{cases} i = e^{+\frac{i\pi}{2}}, & \text{for } \omega < 0 \\ 0, & \text{for } \omega = 0 \\ -i = e^{-\frac{i\pi}{2}}, & \text{for } \omega > 0 \end{cases}$$

More details available at: Hilbert Transform

Where is it used?

Hilbert Transform has application in Quadrature processing : The term quadrature processing refers to dealing with In-phase, Quadrature-phase signals, i.e I,Q signals. I,Q signals are used for modulation & demodulation techniques. Quadrature modulation/demodulation offers many advantages over conventional approaches which can be discussed in another post.

For example, SSB demodulation by phasing method uses Hilbert transform to phase shift the incoming modulated signal to -90°.

Hilbert Transform

A Hilbert transform is essentially a ±90 degree phase shifter

Hence, we define the Hilbert transformer in the frequency domain, with the frequency response function as : $H(f) = -j\ sgn(f)$.

$$sgn(f) = \begin{cases} +1 & \text{for } f>0 \\ 0 & \text{for } f=0 \\ -1 & \text{for } f<0 \end{cases}$$

So, hilbert transform is the convolution of the function x(t) with $1/\pi t$ in time domain which in frequency domain turns out to be multiplication of H(f) and X (f), where X(f) is fourier transform of x(t).

What is Fourier Transform?

The **Fourier transform**, is a mathematical transformation employed to transform signals between time (or spatial) domain and frequency domain, which has many applications in physics and engineering. The new function is then known as the **Fourier transform** or the frequency spectrum of the function *f*. The Fourier transform is also a reversible operation.

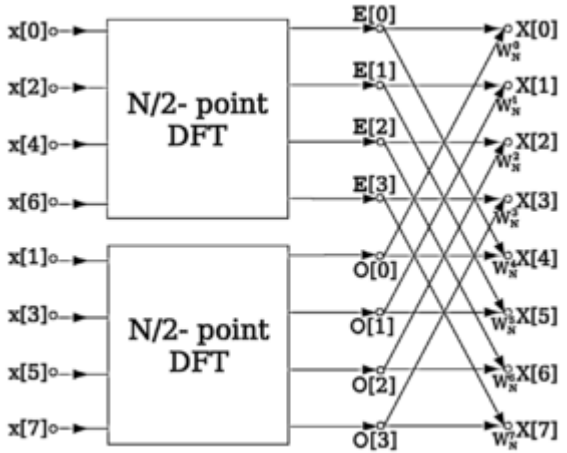
Definition:

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i x \xi}\, dx$$

For more details : Fourier Transform

How to find Fourier Transform?

One way is Cooley–Tukey FFT algorithm which is the most common fast Fourier transform (FFT) algorithm. It re-expresses the discrete Fourier transform (DFT) of an arbitrary composite size $N = N_1 N_2$ in terms of smaller DFTs of sizes N_1 and N_2, recursively, in order to reduce the computation time to $O(N \log N)$ for highly-composite N .



Cooley-Tukey FFT algorithm

A **radix-2** decimation-in-time (**DIT**) FFT is the simplest and most common form of the Cooley–Tukey algorithm.

Radix-2 DIT divides a DFT of size N into two interleaved DFTs (hence the name "radix-2") of size $N/2$ with each recursive stage.

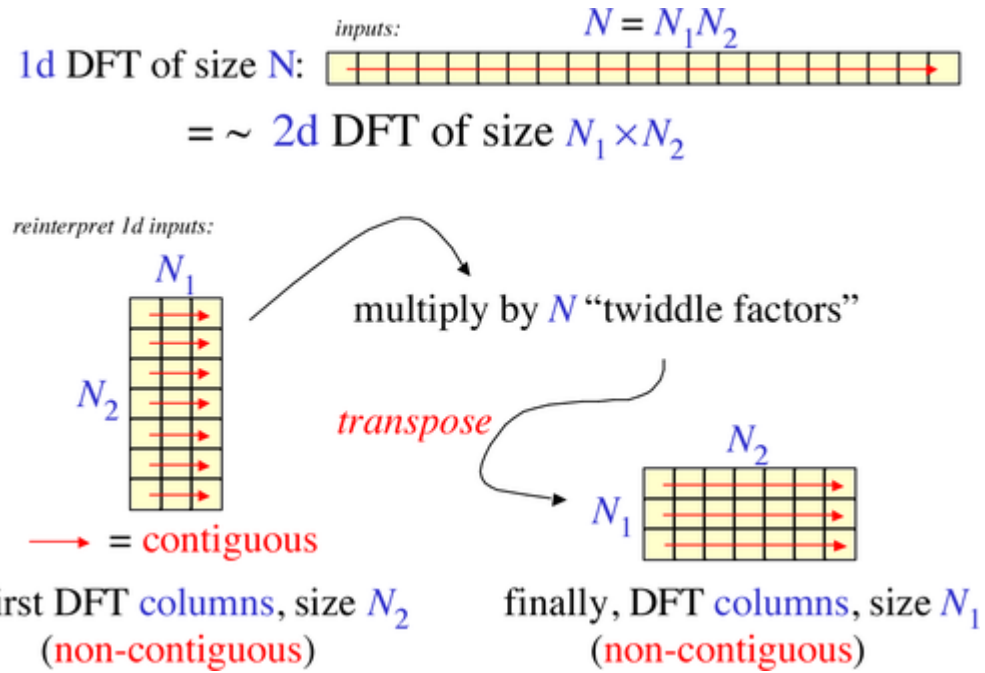
DFT of composite size N= $N_1 N_2$ is expressed as:

- Perform N_1 DFTs of size N_2.
- Multiply by complex roots of unity called twiddle factors.
- Perform N_2 DFTs of size N_1.

For more information : Cooley-Turkey FFT algorithm

Cooley-Tukey FFT algorithm

The basic step of the Cooley–Tukey FFT for general factorizations can be viewed as re-interpreting a 1d DFT as something like a 2d DFT. The 1d input array of length $N = N_1 N_2$ is reinterpreted as a 2d $N_1 \times N_2$ matrix stored in column-major order. One performs smaller 1d DFTs along the N_2direction (the non-contiguous direction), then multiplies by phase factors (twiddle factors), and finally performs 1d DFTs along the N_1 direction. The transposition step can be performed in the middle, as shown here, or at the beginning or end. This is done recursively for the smaller transforms



How to realize Hilbert Transform?

There are many approaches to realize hilbert transform on hardware, what is described here is one of the easiest to understand:

- Find Fourier Transform(FFT) of the function.
- Multiply the FT by exp(-j*π/2)
- Apply inverse fourier transform(IFFT) of the function obtained.

And what you get is hilbert transform of the input function.

Easy, isn’t it?

Okay, What's next?

There are several other approaches to solve this problem statement.You may follow any of them. Finally, judging will be done on the criteria mentioned on the website.

Yes, there are extra points for any extra features, you can think of, where hilbert transform can be applied.

All you have to do is just to keep calm and apply hilbert transform.