Fast Implementation Of Matched Filter With Less Power And Less Delay

Deepak shankhala 1
1 Electronics and communication, Rajasthan University
Nasirabad, Ajmer 305601, India

Abstract

Less time and less delay is required for fast processing for filter **.**the one challenge in processing of image we design a match filter based algorithm. That is resilient to variation in position determination .the second challenge is to fast process of filter. The match filter work to identify the passion of beam result and compared to that obtaining using the centroiding technique .this process required extra processing time for each beam. So for fast implementation processing and less delay time we explore the possibility of using a field programming logic array and parallels combination of each delay block to speed up this computation. The second objective can achieve by the parallel hardware used that provides significant performance improvement over processing.

This paper describes the development of match filter with less delay and fast processing .

Key word :Match filter, FPGA, parallel hardware, fast processing,

Introduction

For laser beam images, centralizing is an acceptable technique for determining the beam position. However, some beam images may exhibit significant intensity variation or other distortions which makes such an approach susceptible to high position uncertainty; in these cases, correlation or matched filtering results in excellent stability even in the presence of fluctuating intensity . Matched filtering using simple templates can achieve fairly stable position detection despite a wide range of intensity and beam quality variations. However, simple templates may not always lead to sufficiently accurate results.

This work demonstrates a template design that yields more accurate results for good beam quality than could be obtained using the simple template, although at the expense of extra processing time required for template creation.

LITERATURE SURVEY

Many of the template-based beam alignment operations are based on 2-D FFT operations which in turn are based on the 1-D FFT. The parallelism inherent in the FFT algorithm allows the hardware implementation to deliver a significant performance improvement over software implementations running on conventional processors. FFT operations allow achieving higher speed by pipelining computations in an FPGA processor. In this study, an FPGA hardware implementation was developed and compared with optimized C and Matlab software implementations. Any compiled Matlab or compiled IDL code which is currently being used in NIF, will only perform as good as a C code but not better. The implementations were tested with a variety of template images. When using 32 template images, the FPGA provided a speedup of about 253 times over the fastest software implementation examined on a 2.0 GHz AMD Opteron core. This FPGA implementation builds on the preliminary designs presented in where speedup factors of only 6 to 20 were achieved. Currently, NIF's 192 beams are aligned in approximately 12 minutes for shot cycles lasting from 4 to 8 hours using compiled IDL algorithms. A faster approach, such as FPGA hardware, will be very useful for lasers requiring continuous alignment operation.

HOW THE MATHEMATICALLY DESIGN MATCH FILTER

.1 Models for matched filtering

Matched filtering is a process for detecting a known piece of signal or wavelet that is embedded in Noise The filter will maximize the signal to noise ratio (SNR) of the sign being detected with aspect to the noise. Consider the model in Figure 2.1 where the input signal is *s(t)* and the noise, *n(t)*. The objective is to design a filter, *h(t)*, that maximizes the SNR of the output, *y(t)*.

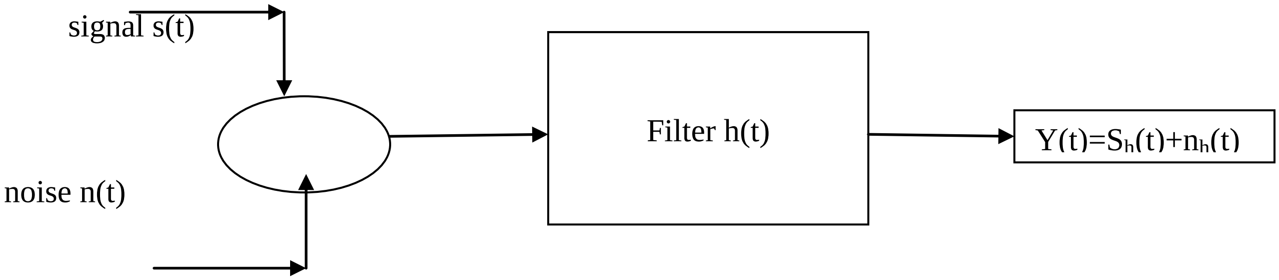


Figure 2.1 Model for matched filtering

If the input signal, *s(t)*, is a wavelet, *w(t)*, and *n(t)* is white noise, then matched filter theory states the maximum SNR at the output will occur when the filter has an impulse response that is the time- reverse of the input wavelet. Note that the convolution of the time-reversed wavelet is identical to cross-correlation of the wavelet with the wavelet (autocorrelation) in the input signal. When the

Wavelet is of length, *T*, then the matched filter is defined by:

$$h(t)= w[T-t]$$

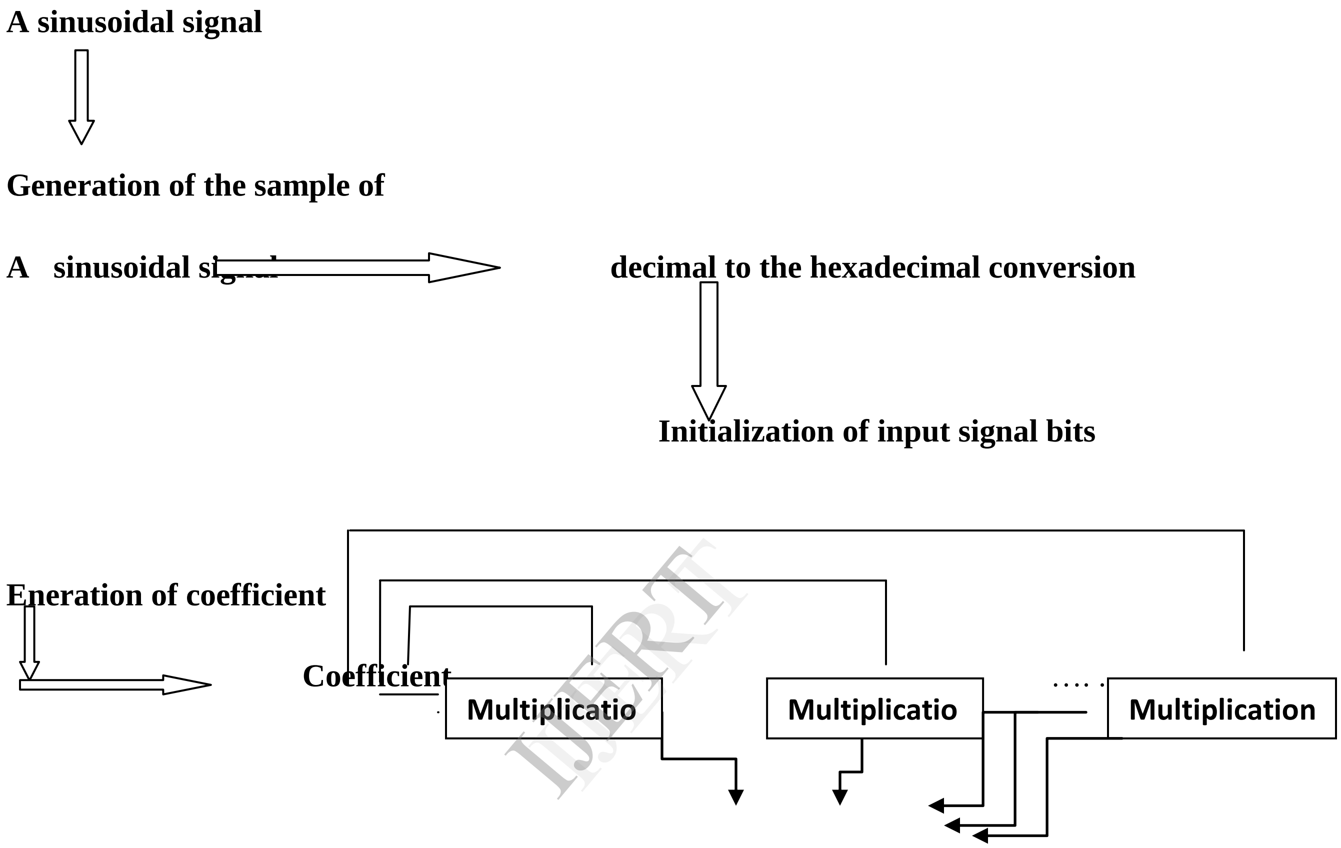
This result is derived in many signal processing texts such as Ziemer and Tranter (1988) and Lathi (1968), and will not be derived in this paper. Essentially, the least squares principle is used to maximize the output signal energy with respect to the output noise energy. The matched filter improves the SNR by reducing the noises spectral bandwidth to that of the wavelet, and in addition, reduces the noise within the wavelet's bandwidth by the shape of the wavelet's spectrum. The duration of the wavelet can be small, as used in radar or sonar, and it can be much larger when used with the vibroseis source with the acquisition of seismic data. Other applications may involve the detection of weak signals from satellite transmissions, or the detection of military equipment from visual images. In addition, Kirchhoff migration is a form of 2D or 3D matched filtering that estimates the location, size, and shape of scattered or diffraction energy.

BACKGROUND

The matched filtering technique utilizes a given object with a known position as a template to find the position of a second object by detecting its position in the correlation domain. The classical matched filter (CMF) and its variation phase only filter (POF) are popular methods for detecting the presence of an object in the presence of noise and distortions..

Flow Chart For Coding

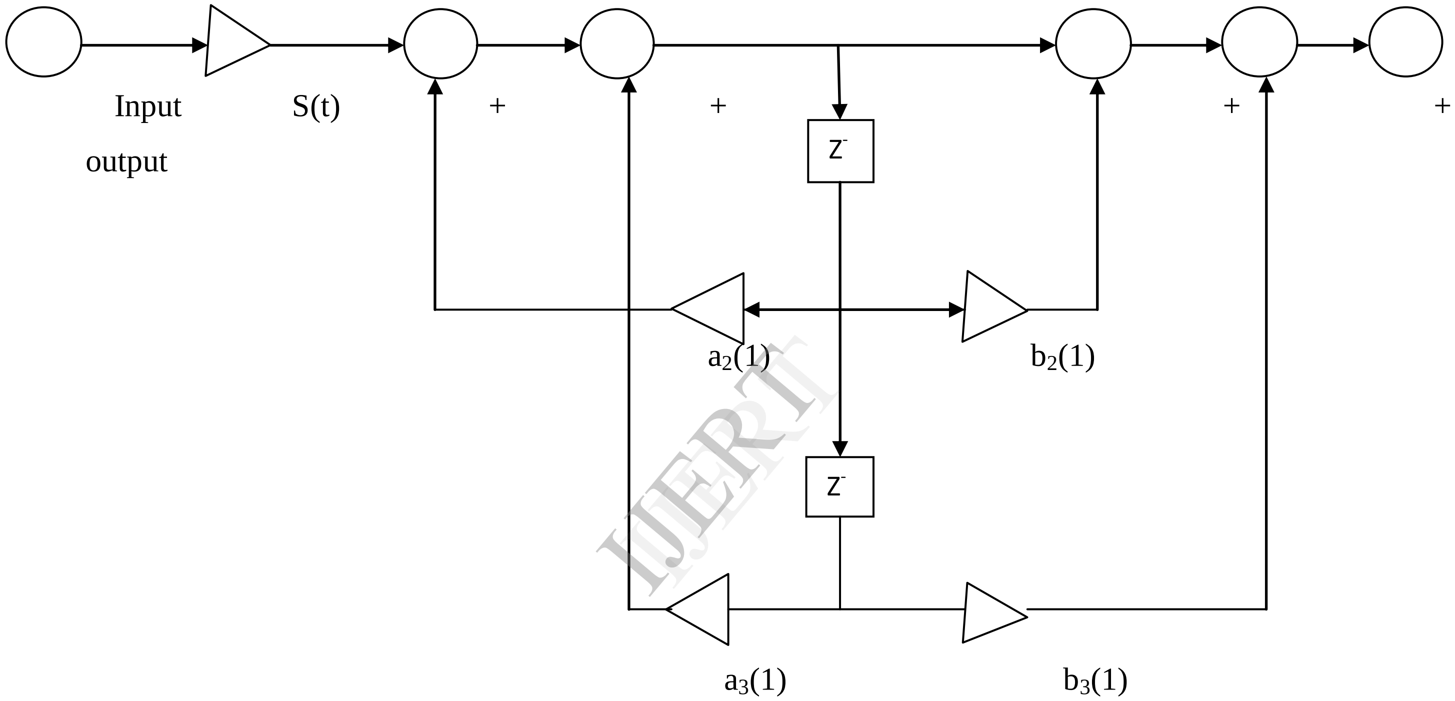
The flow show how you can implement the logic on kit .for less delay



Hardware design

The most computationally intensive portion of the image processing is two-dimensional image correlation. For continuous, high-performance alignment operation such as may be required in a laser inertial confinement fusion power plant, faster methods of beam alignment will be necessary . One advantage of the FFT-based correlation is significant parallelism inherent in the computations, thus enabling potential for greater hardware acceleration. We evaluated hardware acceleration by implementing the image correlation computations on an FPGA. The test system utilized was a Cray XD1 reconfigureurable supercomputer using an architecture based on AMD Opteron processing cores (2GHz) and Xilinx Virtex II Pro FPGAs. Data communication in the system is maximized by integrating the FPGAs at the operating

Stems level and linking them to AMD Option processors through a high-bandwidth, low- latency interconnect. In this system, only one FPGA and AMD core was utilized for the testing since the objective was to compare the performance with a single core CPU and scaled as needed. The AMD core sends images to be processed to the FPGA and receives back the location and peak value in the correlation output.



Block diagram of match filter

Hardware configuration

:

Pre-Phase: The pre-phase consists of applying a match filter to the input image *g(x, y)* to detect the edges. The time for this stage is only seen once because it is overlapped with the computation of the input image *g(x, y)* with the various filters images.

Phase 1: The first one-dimensional FFT for Complex Fourier transform represented is computed. These two computations can be carried out in parallel. The inputs to this phase are unsigned 8 bit values. Since an 8 bit FFT unit would treat the inputs as signed values, a larger bit width FFT unit is needed. Therefore a 12 bit FFT unit is used in the first phase. The first phase 12 bit FFT outputs are stored in buffers labeled mb0 and mb1 exiting Phase 1.

***Phase 2*:** The second one-dimensional FFT to complete the Complex Fourier transform represented is computed. As the maximum output value for Phase 1 is 14 bit, a 16 bit FFT unit is used for the second phase. Also part of is evaluated. Here the output of conjugated and multiplied by the output. An FFT shift operation is executed in parallel with the multiplication in order to center the image. The 40 bit output is stored in a buffer.

Phase 3: The first one-dimensional FFT for the inverse FFT is evaluated. Since the inverse FFT is implemented with two 24-bit forward FFT units, they use only the most significant 24 bits of the inputs. This introduces round-off error as the computations take place in the integer domain.

Phase 4: The second one-dimensional FFT for the inverse FFT is Eq. (5) is evaluated here. Pipelined computation of the location of the peak in the output (C_{CMF}) is also determined. The absolute value of each location is computed and then compared against previously generated values to determine the peak location. The coordinates and amplitude of the peak along with the amplitude of the four surrounding locations are stored and returned to the processor. The template where the maximum has occurred among the submitted templates to the FPGA is also returned to the processor.

Hardware performance

The system above was implemented on a Xilinx Virtex II Pro FPGA (part number XCVP50) on a Cray XD1. The FPGA synthesized system ran at 160 MHz. FPGAs contain a certain amount of logic (AND, OR, etc.) and memory (block RAM) on chip. Any design is converted to a circuit that is programmed on the FPGA. Our circuit used 69% of the logic and 75% of the memory on the FPGA. The algorithm was also implemented in Matlab and in C.

The latter was developed because it would provide a fair comparison of the FPGA against a software implementation. The C implementation is more optimized than Matlab.

Macro Statistics use component

ITEM	NEEDED
Multiplier	81
16*16 bit multiplier	81
Adders/Subtractors	80
32-bit adder	40
39-bit adder	40
# Registers	1334
Flip-Flops	1334
# Multiplexers	40
38-bit 4-to-1 multiplexer	40

Device utilization summary:

ITEM	NEEDED
Selected Device	**4vfx40ff672-12**
Number of Slices	5265 out of 18624 28%
Number of Slice Flip Flops	1367 out of 37248 3%
Number of 4 input LUTs	9903 out of 37248 26%
Number of IOs	57
Number of bonded IOBs	57 out of 352 16%
Number of GCLKs:	1 out of 32 3%
Number of DSP48s	48 out of 48 100

Conclusion

This paper has discussed an effective method for designing FIR filter and match filter of isolated less power consume and less delay time. It presents a parallel designing of filter for image recognition recent years there has been a steady movement towards the development of image recognigation technologies to replace or enhance text input called as have Mobile, video Search Applications. Recently NASA is working search applications. Future work can include improving the recognition filter design of the individual image reorganization by combining the multiple classifiers. Matched filters are designed to extract the maximum SNR of a signal that is buried in noise

References

1. Fast Implementation of Matched Filter Based Automatic Alignment Image Processing, A. A. S. Awwal, K. Rice, T. Taha ,LLNL JRNL 4028821,April 9, 2008,

2 .K. C. Wilhlelmsen, A. A. S. Awwal, S. W. Ferguson, B. Horowitz, V. J. Miller Kamm, C. A. Reynolds October 5, 2007, International Conference on Accelerator and Large, Knoxville, TN, United States Experimental Physics Control Systems, October 5, 2007

3. Implementation of Accelerated Beam-Specific Matched-Filter-Based OPTICAL ALIGNMENT A. A. S. Awwal, K. L. Rice, T. M. Taha ,February 9, 2009

4. A. Awwal et al., “Uncertainty Detection for NIF Normal Pointing Images,” in Optics and Photonics for Information Processing, Proc. SPIE Vol. 6695,66950R (Sep. 20, 2007).

5. Introduction to matched filters, John C. Bancroft

6 Ziemer, R.E., and Tranter, W.H., 1988, Principles of Communications, John Wiley and Sons, pages 465-468.