

Implementation of Transient Signal Detection Algorithms on FPGA

Hamid GholamHosseini
School of Engineering
Auckland University of Technology
Private Bag 92006
Auckland 1142, New Zealand

Kang Li
School of Engineering
Auckland University of Technology
Private Bag 92006
Auckland 1142, New Zealand

ABSTRACT

Radio transient signals are non-periodic and discrete obtained from high energy physical processes in space. Most challenging issues in transient signal detection are the speed and accuracy with which a signal can be detected. Cumulative Sum (CUSUM) algorithm has been employed in this paper for transient signal detection and proved to be capable of meeting the necessary requirements. However, as ordinary software-based programs are unable to handle large scale sample of signals, the current research focuses on implementing the CUSUM algorithm on Field Programmable Gate Array (FPGA) which is a specific integrated circuit within the field of semi-customized circuits that can greatly enhance the speed of detection and analysis. Therefore a FPGA-based system was devoted to implement the algorithms for an efficient transient signal detection. A detection speed of 64 ns per sample set was achieved via implementation the algorithm on an Altera Cyclone IV device with a clock speed of 50 MHz. The analyzed result shows the power consumption of the FPGA based system can be reduced to 136.75 mW.

Keywords

Transient Signal Processing, CUSUM Algorithm, FPGA Implementation

1. INTRODUCTION

Radio transient signals generate from high energy physical processes in space, such as solar flares, supernovae, pulsars, quasars and active galaxies. Other speculations include evaporating black holes, colliding neutron stars and a number of unknown events. The detection of radio transient is a challenge due to their short and non-periodic nature, as well as the high risk of misdetection. It requires the backend of radio telescopes to be equipped with the appropriate hardware and software. Generally, a de-dispersion procedure is used to improve detectability and test the property of the signal. However, due to the large scale of the signal from outer space, the computational demands of this method appear insufficiently robust. Therefore, a new improved algorithm and method for detecting transient signals is developed.

Traditionally most of the developed solutions have been based on software programs targeted for general purpose processors. The shortfalls of these platforms are obvious. For example, these platforms are usually constrained by a fixed number of processors, a limited operating speed and a fixed bandwidth, and are characterized by high power consumption. Most importantly, not all the resources on such platforms are used for transient signal operation, with some parts of the resources consumed by the operating system and software setup. The

platform would be used to optimize only transient signal operation, thereby saving on the cost of other unnecessary

components. In addition, the power consumption of the customized platform would be much lower than a general purpose computer. The most common technologies available to achieve this result are the Application Specific Integrated Circuit (ASIC), and the Field Program Gate Array (FPGA). ASIC is an integrated circuit customized to perform a certain task. The advantages of ASIC include low power consumption, high operating frequency and high logical density. The disadvantages are that ASIC systems come with a high design cost and require specialized designer knowledge.

FPGA is a specific integrated circuit within the field of semi-customized circuits that solves the lack of customized circuits and overcomes the existing limitation of gate numbers. The circuit is designed in hardware description language (Verilog or VHDL) allowing easy layout and burn to the FPGA chip. Another advantage of using FPGA is the capability of working as a co-processor for High Performance Computers (HPC). It seems easier and more reasonable to implement algorithms via FPGAs, with the design on FPGAs be able to migrate to HPCs or ASICs in most cases.

2. DETECTION ALGORITHM

Transient signal detection can be considered as a complex stochastic model. Any abnormal signal can affect changes of the model. The aim of detection is to monitor the difference between input signals with the threshold. Currently, there are two ways to monitor those changes; one is from the perspective of signal processing, the other is from a statistical point of view.

Signal processing methods usually transform the sampling signal to a time domain or frequency domain and observe the changes. In [1], Cornel Loana provides an adaptive time-frequency method based on the over-complete wavelet transform concepts, which lead to signal processing on interest frequency bands. This method is based on the fourth order moment, and is applied for each sub-band, in order to establish the optimal weight for each sample. The result obtained proves the capability of the proposed approach to accurately detect a transient signal, when compared with other methods (e.g. Spectrogram or Standard Wavelet Transform) [2]. The author [1], discovered that the commonly used method, discrete wavelet transform (DWT) was not well suited to this kind of signal processing problem. From a mathematical point of view, the DWT is generated by the sampling in time-scale plant of a corresponding continuous wavelet transform (CWT). Despite the fact that there is an

infinite possible discretization of CWT, the terms of DWT is commonly used to refer to that associated with the dyadic sampling lattice [3]. In certain analyses it will cause wavelet orthogonal basis and the use of orthogonal representation will lose the signal characteristics [4]. In order to eliminate this drawback, the key factor is the use of a non-dyadic sampling structure, which in this case is the Over Complete Wavelet Transform (OCWT). However, this approach requires a more complicated data acquisition process and still needs to be improved [5].

The aim of many statistical methods is to discover the characteristics of the sampled signal. Soudlenkov[6] and Fridman [7] proposed the use of Cumulative Sum (CUSUM) algorithm for transient signal detection capable of meeting the necessary requirements. The Mann-Whitney U [8] and the Wilcoxon signed-rank [9] are both non-parametric statistical methods. The Mann-Whitney U is used for assessing which of two independent observations have larger values than the other and is one of the most well-known non-parametric significance tests. The Wilcoxon signed-rank test is used when comparing two related samples, or repeated measurements on a single sample, to assess whether their population means differ (i.e. a paired difference test). This method is based on the assumption that there is no significant difference between the two samples' overall distribution. The main limitation of these methods is that they were originally designed for detecting single point changes. By contrast, the Mann-Kendall [10] and the CUSUM methods are particularly suitable for sequential analysis. Specifically, the Mann-Kendall method is used to measure the association between two measured quantities. It is easy to implement and widely used in the analysis of climate change. CUSUM is a sequential analysis technique typically used for monitoring change detection [10]. It has several advantages including its relative simplicity, a graphical interpretation of results, and the ability to detect unusual patterns. It has been successfully used in fault detection, onset detection, and defect detection in mechanical systems. Both the Mann-Kendall and the CUSUM tests have particular parameters that need to be fixed at design-time [11] in order to allow the test to detect changes. Specifically, the Mann-Kendall test requires setting a level of significance for the test, while the CUSUM test needs to fix the thresholds in order to detect the possible changes in statistical behavior. One of the significant benefits of CUSUM for signal detection is its stability in the presence of regression behavior for signal sampling [10]. The CUSUM test was chosen as the method of implementation for this research due to its high detection accuracy and real time computation features.

CUSUM is a detection procedure proposed by Page (1954) and Lorden (1971). It is a sequential analysis technique in statistical quality control, typically used for monitoring change detection. As its name implies, CUSUM involves the calculation of a cumulative sum (making it "sequential"). In this algorithm, a constant reference value is subtracted from the data collected. This difference is added to the previous difference (the cumulated sum). Usually this average value is referred to as "M." The equation is summed as below:

$$S_n = S_{n+1} + (X_n - M)$$

The CUSUM test requires a reference threshold value h . When the value of S_n exceeds a certain threshold, an abrupt change can be detected. Threshold selection is the key point for CUSUM detection. However, there is no fixed method to detect transient signals. The author chose Standard Deviation (SD) method. Standard Deviation is a method

widely used in statistics to measure variability or diversity. It shows the extent of variation offset from the mean or expected value. As it is an important indicator of precision in statistics it is widely used in quality control. The SD of a data set is the square root of its Variance. The equation is:

$$S_n = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{N}}$$

Here the \bar{X} refers to the mean value of the sampling unit. In transient signal detection, to find an overall SD is unrealistic, however, one can choose a certain amount of samples and use the SD of the samples as the threshold. In this design, two times the SD of each sample set is selected the sample set threshold. In Normal Distribution, one SD stands for 68% of overall values and two SD are representative of 95% of overall values. In transient signal detection, if the SD of two neighboring samples extends over two times the SD this indicates that one of the samples is abnormal.

The choice of the size of sample set is important, because it's related to detection accuracy. The standard error equation below expresses the relation between sampling size and standard error.

$$\sigma = \frac{1}{\sqrt{N}} \times S_n$$

This equation reflects the degree of dispersion of samples. The smaller of standard error means the samples close to overall average value, otherwise the samples appear more discrete. Obviously, more samples are chosen in each sample set, more closer to overall SD. In practical, sample set normally choose around 50. Here, the author chooses 32 samples as a sample set in design. The main reason is to reduce the incubation period for sampling data processing.

3. DESIGN STRUCTURES

The CUSUM core comprised of several modules including FIFO, Serial to Parallel, Standard deviation and CUSUM algorithm modules. Here, asynchronous FIFO was adopted because CUSUM module may work under different sampling frequency and processing frequency. Asynchronous FIFO can adapt read and write operation under different clock frequency or in same frequency but different clock phase. For high-speed asynchronous FIFO, address control should be handled carefully. From the perspective of logic gate design, the simple "add" operation is complicated as it involves the "carry" and "flip" operations on the counter and these operations easily generate glitches in a high-speed circuit [12]. Therefore, this module employs Gray code in design. With gray code, only one bit changes state from one position to another. To determine the FIFO status, the read address (read_addr) should be first converted to gray code (read_gray). Then the write clock domain is used to synchronize this read address (rag_wt_syn) and convert this address to binary code. Finally, the current write address (write_addr_pl) is allowed to delay for one clock cycle, with the difference between read address (rag_wt_syn) and write address (write_addr_pl) the status of this FIFO. The aim of delaying the write address one clock cycle is to ensure it synchronizes with the read address, because the read address takes one clock cycle to convert to gray code.

The main function of Serial to Parallel module was to parallelize the serial input data from the FIFO module and calculate the average of these samples. A 5-bit counter is used to control the amount of input data. Once completed, the entire data of this interval, plus its mean value, is output in

parallel. The latency of each output interval is only one clock cycle. Standard Deviation module calculated the SD as a threshold for CUSUM algorithm. The standard equation can be split into two parts: cumulative operation and square root operation. Cumulative operation is completed here by parameterized ALTMUTI_ADD megafunction. This function has four groups of “MUTI” port and each port contains two groups of input. Each input can be expressed as the difference between the sample and the mean value. A square calculation can be done by implementing the same input for each port. Figure 1 shows one group of parameterized ALTMUTI_ADD functions.

```
MUTI_ADD_BASE MUTI_ADD_BASE_0
(
    .clock0(clk),
    .dataa_0(idin_0 - mean), .dataa_1(idin_1 - mean),
    .dataa_2(idin_2 - mean), .dataa_3(idin_3 - mean),
    .datab_0(idin_0 - mean), .datab_1(idin_1 - mean),
    .datab_2(idin_2 - mean), .datab_3(idin_3 - mean),
    .result(MUTI_ADDR_OUT_0)
);
```

Fig 1: MUTI_ADD_BASE Module

Each “MUTI_ADD_BASE” module function can complete four groups of input data. Therefore, parallelized eight MUTI_ADD_BASE modules can process one sampling interval data. This approach reduces the calculation time by consuming more resources and is a kind of implementation of the space for speed method. The output from each MUTI_ADD_BASE module is added up and divided by the total numbers. It is then sent to the parameterized SQRT megafunction to calculate get the SD. Pipeline design is implemented here to minimize the clock consumption by “add” operation. It only takes three clock cycles to add eight outputs together. Division calculation is achieved via a shift operation.

In this case, a division of thirty-two can be achieved through shifting five bits left. CUSUM algorithm module is to detect abnormal signals in each sample set. This module is used to cumulate the difference between sampled signals with their corresponding mean value. If the cumulative is more than twice the standard deviation that means the sampled signal is an abnormal signal. Then output this signal for further processing. Otherwise, the output port will be pull up to high resistance status. As the sampling signal is a sign-based signal, the cumulative value may appear as a negative number. Therefore this module should have the ability to detect either positive or negative abnormal signals. Therefore, “2’s complement” method was implemented here, to setup negative threshold.

4. MODULE ASSEMBLE

Modules assembly is the key point in digital design. Unlike C or other high level languages which all functions are executed sequentially, Verilog modules all run in parallel. This means, if stitching these modules simply, they cannot achieve the expected operating results. However, most of the IP cores supported by Quartus are multi-functional and do not provide communication signals such as “Start,” and “Done,” as these cores do not contain sub module, functional module and control module. They are therefore only employed in many “always @” statements inside modules to achieve desired results, which increases the difficulty for sequential operations. There are two ways to solve this problem. The first way is to rewrite two of the IP cores and add

communication signals. Another way of approaching the issue is to consider a parallel operation instead of a sequential operation. From the perspective of saving development time, the author decided to adopt the second method. This method necessitates the precise knowledge of the execution time for each module in order to build the pipeline structure via timing sequential. The results obtained by verification each module shows there are 33 clock cycles consumed by Serial to Parallel module, 9 clock cycles used by Standard Deviation module and another 33 clock cycles spent by the CUSUM algorithm module. Obviously, the running times of these three modules are not equal, meaning that they cannot build using the pipeline structure directly, as it will cause a data hazard when processing old data as new data comes in. Therefore, total 24 clock cycle latency was added on Standard Deviation module. That means although the computing time only consumes 9 clock cycles, the results are being latched until the 33rd clock cycle. Also a Delay module was added in this design. The role of this module is to obtain the output from Serial to Parallel module, delay 33 clock cycles to synchronous with Standard Deviation module and then passes to CUSUM algorithm module. By using this way, this design can achieve pipeline structure and seamless connection for input/output data. Figure 2 shows the pipeline structure of these modules.

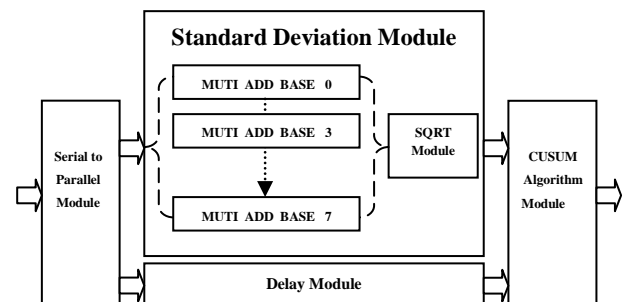


Fig 2: Structure of Design Module

5. VERIFICATION AND EVALUATION

The verification has been done by ModelSim. ModelSim is a powerful HDL simulation tool that allows simulating the inputs of the modules and viewing both output and internal signals. From internal registers Serial to Parallel module output parallelized data can be seen, with the mean value at the 33rd clock cycle. Here, the outputs are separated into two groups, one group to pass to Standard Deviation module and the other group to send to the Delay module to meet the timing requirement. The standard deviation calculation is done at the 65th clock cycle, and then sends the result to the CUSUM algorithm module at the same clock cycle. The internal result register has been integrated to “wire” status; it does not need any extra clock cycles to drive the result to the next module. Delay module passes the data that has been delayed by 33 clock cycles to the CUSUM algorithm module at same time. At the 66th clock cycle, CUSUM algorithm module outputs the detecting result. This continues until the 98th clock cycle to finish the detection of first group of input signals. It can also be seen that the second group of input signals that enter the Serial to Parallel module are waiting for parallelized processing when the first group of input signals are in the standard deviation calculating process. The calculation for the first group and parallelized processing for the second group are completed at same time. This means that at the time that the second group signals enter the CUSUM algorithm module, the first group signals have just passed through. Similarly this operation applies for the following groups. It can be concluded

from the above that the entire latency is 65 clock cycles. This means it will consume 65 clock cycles to process one group of input signals. In this way the pipeline structure can achieve a continuous processing flow, avoid wasting clock cycles, and maximizes processing speed. After observing several groups of sampling data, it was found that the data could achieve seamless transfer without any missing. This module can therefore be considered to meet the design requirements.

The power consumption of the IP core was measured by the PowerPlay early power estimator tool which support by Altera. The power consumption is composed of static and dynamic power. Static power is the power consumed by leakage current. Dynamic power is the amount of power the device consumes when it is actively operating. Power analyze was done by implementing 1, 2, 4, 8 cores respectively. Table 1 below gives the results.

Table 1. Power consumption results

Number of Cores	Static Power (Mw)	Total Power (Mw)
1	98.9	136.75
2	99.36	144.8
4	100.25	159.7
8	436.94	484.05

One can see that the power consumption tracks in an approximately linear fashion with the increase in the number of cores. This is because the resource utilization is directly proportional to the number of cores. Compared with computer based transient signal detection, FPGA based detection consumes less power as it doesn't need to waste power on software implementation.

6. DISCUSSION AND CONCLUSION

There are still rooms for improvements to this system in order to achieve processing several TByte data per day. These improvements can be achieved via multiprocessor operation and multi-core operation. Multiprocessor architecture can enhance processing speed. The SOPC Builder was observed to allow users to add custom instructions to the Avalon Bus and build their own system via the NIOS II processor. The number of processors in a multiprocessor system was shown to be scalable, with processors able to be easily added or removed from the multiprocessor system. Avalon Bus can easily control the on-chip memories which store the shared data in the detection process. Further, the workload of detection can seamlessly allocated to any number of processors. The resource limitation of extending such a multiprocessor system is mainly associated with the on-chip shared memories and logic resources. Therefore, more processors are able to be easily added to improve the performance if more on chip memory is available. One needs to balance the process efficiency however, with the number of processors. This efficiency ratio is defined as [13]:

$$E = \frac{\text{Execution time using one processor}}{\text{Execution time using multiprocessor} \times \text{number of processors}}$$

If the computational complexity is not sufficient to have all processors running at the same time, adding more processors will decrease the performance and also waste resources. Another issue that needs to be considered is the use of one NIOS II processor to control a multi CUSUM core, or the use of multiprocessors to process one CUSUM core. However selecting the wrong architecture will introduce side effects to the processing speed. Here, using one NIOS II processor to

control multi CUSUM core can be considered as intra-node architecture, while using a multiprocessor to process one CUSUM core can be considered as inter-node architecture. Generally, for small or medium complexity computation, intra-node architecture achieves higher performance than inter-node architecture. This is due to the communication time required within inter-node architecture, which slows down the processing speed. If the computational complexity is increased multiprocessor architecture will deliver improved performance, so there needs to balance these two forms of computing architecture in future work.

7. REFERENCES

- [1] Loana Cornel, Andre Quinquis, Transient Signal Detection Using Overcomplete Wavelet Transform and High-Order Statistics. Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03), 2003. 6: p. VI - 449-52 vol.6
- [2] A. K. Jayaprakasam, Cooperative Robust Sequential Detection Algorithms for Spectrum Sensing in Cognitive Radio. Ultra Modern Telecommunications & Workshops, 2009: p. 1 – 8.
- [3] S. Zarrin, T.J. Lim, Cooperative Quickest Spectrum Sensing in Cognitive Radio with Unknown Parameters. Dept. of Electr. & Comput. Eng. 2009, Univ. of Toronto: Canada p. 6.
- [4] A. Theolis, Computational Signal Processing with Wavelets. 1998: Birkhauser, Boston.
- [5] H. Aghajan, Y. Pati. and Kailath, Transient Signal Detection Using High Resolution Line Detection on Wavelet Transforms, in Signals, Systems and Computers. 1994 USA. p. 1114 - 1118.
- [6] Gene Soudlenkov, Vyacheslav V. Kitaev, Two-staged Algorithm for Dispersed Transient Radio Emission Detection, in Engineering. 2010, Auckland University of Technology: New Zealand p. 13.
- [7] P.A. Fridman (2010) A Method of Detecting Radio Transients. 2010. 409(2): p. 449-880.
- [8] G. Zeoli and W. Fong, Performance of a Two-Sample Mann-Whitney Nonparametric Detector in a Radar Application. Aerospace and Electronic Systems, IEEE Transactions 2007. AES-7 (Issue:5): p. 951 – 959.
- [9] B. Seyfe and A.R. Sharafat, Signed-rank nonparametric multiuser detection in non-Gaussian channels IEEE Information Theory Society 2005. 51, Issue:4 (4): p. 8.
- [10] C. Roveri and R. Manuel, An adaptive CUSUM-based test for signal change detection in Circuits and Systems, 2006. ISCAS 2006. Proceedings. pp. 4.
- [11] X. H. Kun and S. H. Yang, Fundamental of Cumulative Sum Method and its Application in Measurement Data Processing, in Dept of Measurement and Control. 2000, He Fei University of Technology: He Fei.
- [12] S. Kilts, Advanced FPGA Design: Architecture, Implementation. 2007, Hoboken.
- [13] M. Allen and Wilkinson, Parallel Programming Techniques and Applications Using Networked Workstations and Parallel Computers. 2006.