Parallel Computation based Spectrum Sensing Implementation for Cognitive Radio

Shixian Wang, Hengzhu Liu, Botao Zhang, Lunguo Xie
School of Computer, National University of Defense Technology, Yanwachi str. 47, CN-410073 Changsha, China, phone: +86 731 84517834, e-mail: sxwang@nudt.edu.cn

crossref http://dx.doi.org/10.5755/j01.eee.118.2.1164

Introduction

The past decades witnessed the fast development of the wireless communication technologies and its applications. However, the upgrading demand is contradictory to the limited available radio spectrum resource because of the traditional frequency allocation policy. Cognitive Radio (CR) or Dynamic Spectrum Access (DSA) was proposed to improve the utilization of the assigned but unused radio spectrum [1–3].

The key idea of CR is to utilize the unused radio spectrum assigned to primary users (PUs) via successful spectrum sensing, so that CR nodes will not introduce any interference to primary users during their communication process. Therefore, the ability to sense the spectrum reliably and timely is quite critical to CR.

There are several typical spectrum sensing alternatives, such as matched filter, energy detector and cyclostationary detector. Matched filter requires prior knowledge of the waveforms and channel responses of the primary users, which may not be possible if there is no corporation between primary and secondary users. Energy detector has very simple structure to be implemented, but its performance degrades when noise level is unknown or changing. In addition, energy detector is incapable of differentiating between modulated signals, noise and interference.

On contrast, cyclostationary feature detector has the ability to separate the interested signal from noise and/or interference, which has made it more suitable to detect licensed users in a CR environment. Furthermore, cyclostationary feature detection has been used in multiple narrowband signals difference of arrival (DOA) estimation, weak spread-spectrum communication signal identification, and radar signal parameter estimation [4]. However, high computational complexity of cyclostationary feature detection limits its usage as a real-time signal analysis tool.

Nevertheless, cyclostationary feature detection is still being regarded as a competitive spectrum sensing method because of its special capabilities. Cyclostationary feature detector is implemented and tested on the BEE2 platform for spectrum sensing in [5], where the results show that feature detectors are more robust than conventional energy detectors. An FPGA implementation of a cyclostationary feature detector is proposed in [6], which has an improved detection performance achieved by decimation of the cyclic spectrum. These works implement the cyclostationary feature detectors on the FPGA platforms, only with the aim of CR prototype system verification.

To reduce the computational complexity of cyclostationary feature detection, the computational efficient FFT accumulation method (FAM) is proposed in [7]. Parallel algorithm and algorithm parallel implementation are feasible ways to attack the computational complexity of many applications. It has been proved that the parallel FAM algorithm can be 7.6 times faster than the serial FAM algorithm on the CELL BE parallel computer platform [8]. The CELL BE is a powerful computing platform, but it was never designed to be a mobile solution [9]. Mapping the original cyclostationary feature detection algorithm onto the Montium cores is proposed in [10]. Although the Montium cores provide both flexibility and power efficiency, the implementation in [10] cannot guarantee real-time spectrum sensing. Therefore, a more suitable platform and more efficient parallel algorithm still need to be studied for the spectrum sensing task of CR.

CR is an intelligent radio based on the software defined radio (SDR) platform, which has powerful computation capability. Therefore, a SDR processor will be a suitable platform for CR spectrum sensing. A novel SDR baseband processor for the next generation wireless communication, named GAEA, has been proposed in [11], with the peak computing capability of 43.75 GOPS and power consumption of 481 mW. Due to its satisfactory performance, it will be used as the implementation platform for CR spectrum sensing in this paper.

The present paper firstly introduces the system model
of spectrum sensing for CR. Then, a detailed analysis of cyclostationary signal characteristics and an enhanced computational efficient parallel FAM algorithm for cyclostationary feature detection are presented. The parallel FAM algorithm is implemented on a SDR processor, which makes this method more feasible for CR spectrum sensing. Moreover, this paper proposes the parallel implementation method, experiment environment and experiment results, which is useful for other cyclostationary feature detection applications.

Cognitive radio spectrum sensing system model

Cognitive radio spectrum sensing problem can be considered as a kind of signal detection problem which has been investigated for many years. According to the signal detection theory [12], the system model for spectrum sensing of CR can be expressed as follows.

Assume \( x(n) = s(n) + w(n) \) stand for the received signal of a primary user, passing through a channel with path loss, multipath fading and time dispersions, where \( s(n) \) is the possible primary user’s signal and \( w(n) \) is the noise. Spectrum sensing can be considered as a binary hypothesis testing problem with:

\[
H_0 : x(n) = w(n),
\]

\[
H_1 : x(n) = s(n) + w(n),
\]

where \( H_0 \) represents the hypothesis that the primary user’s signal is absent, \( H_1 \) represents the hypothesis that the primary user’s signal is present and \( n = 1, 2, \ldots, N \) are indices of \( N \) signal samples.

Spectrum detection performance can be characterized by probability of miss detection \( (P_{md}) \) and probability of false alarm \( (P_f) \), which are all important parameters. A miss detection occurs when a busy channel is detected as idle, which means at hypothesis \( H_0 \), the probability of the detector having not detected the signal. And a false alarm occurs if an idle channel is detected as busy, which means at hypothesis \( H_0 \), the probability of the detector having detected the signal. That is, the following definitions hold:

\[
P_{md} = \Pr(H_0 | H_1),
\]

\[
P_f = \Pr(H_1 | H_0)
\]

and the probability of detection, \( P_d \), is defined as

\[
P_d = \Pr(H_1 | H_0) = 1 - P_{md}.
\]

Among the evaluation metrics, spectrum sensing time is another important parameter. It should be as short as possible to avoid any interference to PUs when they come back to use their spectrum. According to the first IEEE standard on cognitive radio, IEEE 802.22 standard draft, the sensing time should be not more than 2 seconds [13]. In addition, other essential implementation parameters include the frequency resolution and bandwidth, power and area consumption.

Cyclostationary feature detection and FAM algorithm

Modulated signals have built-in periodicity. Even though the data is a stationary random process, these modulated signals are characterized as cyclostationary, since their mean and autocorrelation exhibit periodicity. This can be used for the detection of a random signal with a particular type in a background of noise and other modulated signals. A signal process \( x(t) \) is said to be second order cyclostationary if its mean and autocorrelation are periodic with a period \( T_0 \), i.e.:  

\[
M_x(t+T_0) = M_x(t),
\]

\[
R_x(t+T_0, \tau) = R_x(t, \tau).
\]

Then, the periodic function \( R_x(t, \tau) \) can be further expressed as follows

\[
R_x(t, \tau) = \sum_{n=-\infty}^{\infty} R_{x_n}^{T_0}(\tau) e^{-j2\pi f_n T_0} dt = \lim_{N \to \infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x(t+n \cdot T_0)x^*(t+n \cdot T_0 - \tau),
\]

where \( * \) denotes the complex conjugate operation and the Fourier coefficients in equation (5) can be expanded as

\[
R_{\alpha}^{T_0}(\tau) = \frac{1}{T_0} \int_{-T/2}^{T/2} x(t + \tau/2)x^*(t - \tau/2)e^{-j2\pi \alpha \frac{T_0}{T_0}} dt,
\]

which are known as cyclic autocorrelation functions. Let \( \alpha \) represent the frequencies \( \{|n| T_0\}_{n \in Z} \), which is referred to as cycle frequency. The nonzero correlation (second-order periodicity) characteristic of a time series \( x(t) \) exists in the time domain, if the cyclic autocorrelation function is not identically zero. That is, the signal \( x(t) \) is said to be cyclostationary if \( R_{\alpha}^{T_0}(t) \) does not equal zero at some time delay \( \tau \) (any real number) and cycle frequency \( \alpha \neq 0 \).

Then, the spectral correlation density (SCD), or cyclic spectral density, can be obtained from the Fourier transform of the cyclic autocorrelation function (6) as

\[
S_{\alpha}^{T_0}(f) = \int_{-T/2}^{T/2} R_{\alpha}^{T_0}(\tau)e^{-j2\pi f \tau} d\tau = \lim_{T \to \infty} \frac{1}{T} X_f(f + \alpha/2)X^*_f(f - \alpha/2),
\]

where \( \alpha \) is the cycle frequency and

\[
X_f(f) = \int_{-T/2}^{T/2} x(t)e^{-j2\pi ft} dt
\]

is the Fourier transform of the time domain signal \( x(t) \). Power spectral density is a special case of a spectral correlation function for \( \alpha = 0 \). Measurements of (8) and (9) in signal analysis constitute what are referred to as cyclic spectral analysis.

Since the signals being analyzed are defined over a finite time interval \( T \), the cyclic spectral density is only an estimation. Methods to estimate the cyclic spectral density or SCD include time smoothing and frequency smoothing. Time smoothing algorithms are considered to be more computationally efficient for general cyclic
spectral analysis [5]. An estimate of the SCD can be obtained by the time-smoothed cyclic periodogram given by
\[
S^a_x(f) \approx S_{xy}^a(t, f) = \frac{1}{T \Delta f} \int_{-\Delta t/2}^{\Delta t/2} X_{xy}(u, f + \alpha / 2) \overline{X_{xy}(u, f - \alpha / 2)} \, du,
\]
where \( \Delta t \) is the total observation time of the signal. The cycle frequency resolution of the estimation, \( \Delta \alpha \), is determined by \( \Delta \alpha = 1 / \Delta f \). \( T \) is the short-time FFT window length and
\[
X_{xy}(t, f) = \int_{t-\Delta t/2}^{t+\Delta t/2} x(n) e^{-j2\pi fn} \, dn
\]
is the sliding short-time Fourier transform. The spectral components generated by each short-time Fourier transform have a resolution of \( \Delta f = 1 / T \). In addition, there is an overlap factor, denoted by \( L \), between each short-time FFT.

For a reliable and accurate estimation of \( S^a_x(f) \) from (7) for any given \( t, f \) and \( \Delta f \), the observation time \( (\Delta t) \) must greatly exceed the time window \( (T) \) that is used to compute the spectral components. In order to avoid aliasing and cycle leakage on the estimates, the value of \( L \) is defined as \( L \leq Tw / 4 \).

The FFT accumulation method is a time-smoothing algorithm and proved to be the computationally efficient than frequency-smoothing algorithms. For the discrete SCD, we define the sampled signal
\[
x(n) = x(n - L),
\]
where \( f_s \) indicates the sampling frequency. Furthermore, we assume parameter \( N \) represents the total number of discrete samples within the observation time, and \( N \) represents the number of points within the discrete short-time FFT. Then we get the discrete Fourier transform of \( x(n) \), \( X_N(n, f) \)
\[
X_N(n, f) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi f n / N},
\]
where \( w(n) \) is the data taper window (e.g., Hamming window). Then, the discrete SCD becomes [4, 8]
\[
S_{xy}^a (n, f) = \frac{1}{N} \sum_{n=0}^{N-1} \left[ X_N(n, f + \alpha / 2) \overline{X_N(n, f - \alpha / 2)} \right].
\]

To sum up, a system level computing flowchart of the FFT accumulation method can be described by Fig. 1.

As Fig. 1 depicted, after the input signal channelization function, an array is formed with rows which are \( N \) points long, and each succeeding row’s starting point is \( L \) samples offset in the original sample sequence from the previous row’s starting position. Then a window is applied across each row. The \( N' \)-point FFT results are shifted in frequency in order to obtain complex demodulate sequences. After the complex demodulates are computed, product sequences are formed and Fourier transformed with a \( P \)-point FFT \( (P=N/L) \).

Therefore, the computational efficiency of the algorithm is improved by a factor of \( L \), since only \( N/L \) samples are processed for each point estimate. The cycle frequency resolution of the decimated algorithm is defined as \( \alpha_{res} = f_s / N \), and the frequency resolution is \( f_{res} = f_s / N' \). For a statistically reliable measurement, the time-frequency resolution product \( N / N' \) must satisfy \( N / N' \gg 1 \), and meanwhile, the \( N' \) should be large enough to obtain the desired frequency resolution.

**Parallel FAM algorithm and implementation platform**

In order to implement FAM algorithm described in last section, we relied on the SDR processor for the next generation wireless communication, named GAEA proposed in [11]. GAEA is a shared memory static scheduling multi-core system-on-chip (SoC) based on a novel bus named Software Controlled Time Division Multiplexing (SC-TDM) Bus. Programmers can explore memory level parallelism of applications with proper instructions and bus scheduling algorithms based on SC-TDM bus. Fig. 2 shows the system level architecture of GAEA.

GAEA is mainly comprised of data processing units, Matrix Process Engine (MPE) and a control unit LEON3 [14]. There are 4 MPEs in one GAEA, and sometimes 1-2 coprocessors will be appended to process the ultra complex algorithms, such as LDPC or Turbo decoder. MPEs and coprocessors share the Level 2 (L2) memory through the 64-bit SC-TDM bus.

MPE, which is the kernel of GAEA, adopts hybrid parallel processing scheme to explore instruction-level and data-level parallelism. Fig. 3 shows the MPE architecture. MPE uses one integrated pipeline to support the 4-issue of scalar instructions and SIMD vector instructions. The width of the SIMD vector is 4-cluster or 8-cluster for half word instructions.

FAM algorithm can be categorized as streaming applications because they fit the producer-consumer model and have a high degree of data parallelism in every stage. Based on the algorithm description in the above section,
we propose a parallel FAM algorithm and its mapping method on the GAEA architecture in Fig. 4 for the purpose of decreasing computation time.

![Fig. 2. GAEA architecture](image2)

In every stage of FAM algorithm, the procedure can be parallelized and mapped onto a macro pipeline. As Fig. 4 illustrates, the input data will be distributed to four MPEs by LEON RISC processor in the first step, and the main operations in this step are all data moving. After input channelization procedure, every MPE has the input data of an array of $N \times P/4$, that is, 1/4 of the columns of the channelization array is in one MPE. Then the Windowing, $N$-point FFT and Down Conversion operations are all executed in parallel on the four MPEs. Column multiplication can also be executed in parallel on MPEs, but we should do some data moving operations. Every MPE first transposes its data array and sends it to the L2 memory, and then the conjugate of the transposed array can be got in every MPE and will be sent to the L2 memory. Every MPE will get its new transposed array by the control of LEON. After that, every MPE receives a conjugated array of $P \times N'$ of the newly transposed array, then the column multiplication can be executed in parallel on the four MPEs. Every MPE has $P \times (N')^2/4$ data before the second stage $P$-point FFT. In the last, the $P \times (N')^2$ array is sent to the L2 memory for data analysis.

We have analyzed the parallel FAM mapping method on GAEA of 4 MPEs in task level. The data level parallelism is supported by VL IW (very long instruction word) and SIMD (signal instruction multiple data) instructions. In MPE, the signed fixed-point integer (word) is 40-bit, which is specified for signal processing applications. Data moving between the LEON and the MPEs is done through DMA operations, with the speed of 8 bytes/cycle. Furthermore, the FFT realization is supported by a specific data exchange network of MPE architecture [11].

The spectrum sensing method was implemented on a Lyrtech SDR development platform. The SDR platform consists of two integrated modules: the digital signal processing module and the data conversion module. Simulated signals are created by an Agilent signal generator, which is connected to the data conversion module.

SignalMaster Quad Virtex-4 card contains the digital signal processing module. It is composed by four DSPs of TMS320C6416 from TI and two Virtex-4 XC4LX100 FPGAs from Xilinx. C64x+ DSP is an advanced VLIW core, which can be used to perform complex data processing. The FPGA-DSP clusters of the SignalMaster Quad Virtex-4 are connected to each other through high-bandwidth inter-FPGA RapidCHANNEL RX/TX buses capable of sustained, full-duplex, 8-Gbps raw data exchanges. The data conversion module is equipped with a 125 MSPS 14-bit dual channel ADC and a 500 MSPS 16-bit dual channel interpolating DAC provided by TI.

![Fig. 3. Matrix Process Engine architecture](image3)

![Fig. 4. Parallel FAM algorithm and mapping method on GAEA](image4)

![Fig. 5. The top level model view of the parallel computation based spectrum sensing implementation in Lyrtech integrated development environment](image5)
Fig. 5 illustrates the top level model view of parallel task implementation. The FPGAs in our platform are not large enough to contain four intermediate version MPE cores. Therefore, we divide the parallel tasks into four parts in our experiment which can be executed on FPGA and the DSP resources. Two FPGAs contain two MPE cores, and the other two parts are executed on two DSP cores.

The measurement setup is illustrated in Fig. 6. Detection algorithm was implemented on our platform that was accompanied by a commercial signal generator for measurement purposes. The SignalMaster Quad Virtex-4 can only be used in a cPCI (compactPCI) chassis system. In addition, we use the local cPCI chassis system configuration model, where the necessary software is installed directly on the CPU board of the cPCI chassis system. And the Matlab was used to control the measurements and analyze the results.

Simulations and experiments

We simulated the tasks of 32768 samples on an 8 MHz bandwidth signal with the GAEA cycle accurate simulator. Table 1 gives an overview of the number of processor cycles required for the different tasks.

Table 1. Number of processor cycles of every task on one MPE

<table>
<thead>
<tr>
<th>Task</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialization</td>
<td>131648</td>
</tr>
<tr>
<td>Complex multiplication</td>
<td>4259840</td>
</tr>
<tr>
<td>N'-point FFT</td>
<td>117760</td>
</tr>
<tr>
<td>Data reshuffling</td>
<td>524608</td>
</tr>
<tr>
<td>P-point FFT</td>
<td>18350080</td>
</tr>
<tr>
<td>Final stage DMA</td>
<td>4202496</td>
</tr>
<tr>
<td>Total</td>
<td>27586432</td>
</tr>
</tbody>
</table>

The only serial part of our algorithm is the data moving operations between L2 memory and MPEs’ L1 memory, which will be realized by DMA. The complex multiply included in three parallel operations stages, which are, windowing, down conversion, and column multiplication. And the total number equals \( \frac{N' \cdot P + N' \cdot P + (N')^2 \cdot P}{4} \). Simulations show that a complex multiply requires one cycle because of the SIMD instructions, so the required number of clock cycles equals 4259840. The intermediate version MPE can calculate the 128-point FFT in 460 cycles and the 1024-point FFT in 4480 cycles, including data displacement and bit-reversal instructions for the right results. The total number of cycles for calculating the SCD equal 27586432. GAEA will run at the frequency of 350 MHz and therefore the required time for parallelized cyclostationary feature detection equals 78.8ms.

In our experiment, half of the tasks were executed on the TI TMS320C6416 cores. As MPEs have more function units and more specific instructions than C64x, C64x are slower than MPE for some signal processing application. For example, 1024-point FFT computing in MPE can be three times faster than it is executed in C64x. However, we finished the task of spectrum sensing at about 126ms in this parallel computation based experiment. The time required of the proposed implementation is smaller than the approach proposed in [7] on the same samples, which is 446ms. The CELL BE has high parallel computing ability, but the effective mapping algorithm provides some help in reducing the processing time on our processor. Furthermore, the high level programming will surely decrease CELL BE’s performance.

It has been found that both simulated carrier and BPSK signal under different SNR level can be detected by our implementation. Table 2 lists the detection performance of the implementation, which proves the feasibility of our approach. Furthermore, our SDR processors based implementation method has more advantage in the low power and high performance.

Table 2. The probability of detection at 0.1 false alarm probability of parallel FAM based detector at low SNR values

<table>
<thead>
<tr>
<th>SNR</th>
<th>Carrier</th>
<th>BPSK Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR=-20dB</td>
<td>0.78</td>
<td>0.39</td>
</tr>
<tr>
<td>SNR=-15dB</td>
<td>0.95</td>
<td>0.82</td>
</tr>
<tr>
<td>SNR=-10dB</td>
<td>0.99</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Conclusions

To reduce the computation time of the cyclostationary feature detection, this paper proposes an efficient parallel FAM algorithm on a novel SDR processor named GAEA. Parallelized cyclostationary feature detection implementation for a common parameter set of spectrum sensing (32768 samples) can be finished within approximately 126 ms on our experiment platform, which is shorter than the spectrum sensing time requirement of CR. The high performance and low power SDR processor provides a feasible way for CR spectrum sensing implementation.

Cyclostationary feature detection has been widely used in the signal processing applications, like multiple narrowband signals DOA estimation, weak spectrum communication signal identification, and radar signal parameter estimation. The implementation method proposed in this paper can be used in most of the applications with high performance and low power thanks to the SDR processor. Moreover, the parallel algorithm is expandable and can be mapped to more processing engines for real-time applications.
Acknowledgements

This work was partly supported by the Doctor Program Foundation of Education Ministry of China (No.20094307100009), National Nature Science Foundation of China (No.60970037).

References


Accepted after revision 2011 09 29


The reliably and timely spectrum sensing ability is very critical to cognitive radio. Cyclostationary feature detection has the ability to separate the signal of interest from noise and/or interference, but the computational complexity of cyclic spectral analysis limits its use as a signal analysis tool. To reduce the computational complexity of cyclic spectral analysis, this paper proposes an efficient parallel FFT accumulation method (FAM) algorithm on a novel SDR processor for next generation wireless communication, GAEA. Parallelized cyclostationary feature detection implementation for a common parameter set of spectrum sensing (32768 samples) can be finished within approximately 126 ms on our Lyrtech SDR experiment platform. The algorithm is expandable and can be mapped to more processors to get shorter detection time. This approach is suitable for spectrum sensing of cognitive radio and other cyclostationary feature detection applications. Ill. 6, bibl. 14, tabl. 2 (in English; abstracts in English and Lithuanian).


Gebejimas justi spektros yra labai svarbus kognityviniam radijui. Cycostacionarumą įsitikinti išskirti reikiamą signalą iš triukšmo ir/arba interferencijos, tačiau ciklinės spektrinės analizės skaičiavimų sudėtingumas truko naudoti ją signalams tirti. Siekiant supaprastinti ciklinės spektrinės analizės skaičiavimus, pasiūlytas efektyvus lygiagreitas FFT akumulavimo metodas algoritmą, realizuotas SDR procesoriuje kitos kartos bevielėje komunikacijose. Algoritmas gali būti išplėstas ir pritaikytas daugeliui procesorių norint sumažinti detekavimo trukmę. Il. 6, bibl. 14, lent. 2 (anglų kalba; santraukos anglų ir lietuvių k.).