

ПРОГРАММА ДЕШИФРАЦИИ И АНАЛИЗА РАДИОЛОКАЦИОННОЙ ИНФОРМАЦИИ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ CUDA

Аннотация.

Актуальность и цели. При проведении отладки и испытаний бортовых радиолокационных станций авиационных комплексов дальнего радиолокационного обнаружения применяется технологическая контрольно-записывающая аппаратура, которая обеспечивает запись на накопитель радиолокационного сигнала. Зарегистрированная информация в последующем анализируется на ЭВМ в целях оценки эффективности работы радиолокационного комплекса. Это позволяет сократить сроки разработки и затраты за счет уменьшения количества летных испытаний. В связи с этим разработка новых алгоритмов обработки радиолокационных данных является актуальной. В данной статье описывается новый алгоритм дешифрации зарегистрированной радиолокационной информации с использованием технологии CUDA.

Материалы и методы. Для решения поставленных целей была разработана программа для ЭВМ с применением технологии CUDA. Для тестирования на вход программы подавались тестовые наборы реальных полетных данных.

Результаты. Экспериментально доказано, что разработанное программное обеспечение на основе предложенного алгоритма дешифрации и анализа радиолокационной информации позволяет повысить эффективность обработки данных и дает прирост производительности до 10 % по сравнению с существующим алгоритмом параллельной обработки данных на центральном процессоре.

Выводы. Предложенный алгоритм и разработанная программа нашли применение при создании программного комплекса для авиационных комплексов дальнего радиолокационного обнаружения нового поколения.

Ключевые слова: технология CUDA, радиолокационные данные, многопоточная обработка.

Е. П. Lepilina

RADAR INFORMATION DECODING AND ANALYSIS PROGRAM USING THE CUDA TECHNOLOGY

Abstract.

Background. For debugging and testing of on-board radar aircraft systems Airborne Early Warning (AS AEW) experts apply technological control and recording equipment (CRE) that provides recording of radar signals on a drive. Recorded information is subsequently analyzed by a computer in order to assess the effectiveness of the radar system. This reduces development time and costs by decreasing a number of flight tests. In this regard, development of new algorithms for processing of radar data is relevant. This article describes a new algorithm for decoding recorded radar data using the CUDA technology.

Materials and methods. The program was designed for computers using the CUDA technology to achieve the targets. The program was tested on a set of actual flight data.

Results. It is experimentally proved that the developed software based on the proposed algorithm of decryption and analysis of radar data increases the efficiency of data processing and provides a performance boost of up to 10% compared with the existing algorithms of parallel processing on CPU.

Conclusions. The proposed algorithm and the developed programs have been used in creation of software for a new generation of AS AEW.

Key words: CUDA, radar date, multithreaded.

Введение

При проведении испытаний бортовой радиолокационной станции (РЛС) комплекса дальнего радиолокационного обнаружения (ДРЛО) применяется технологическая контрольно-записывающая аппаратура (КЗА), обеспечивающая регистрацию информации, передаваемой бортовыми РЛС в бортовую информационно-управляющую систему для дальнейшей обработки. Зарегистрированная информация в последующем анализируется на ЭВМ в целях оценки эффективности работы радиолокационного комплекса [1–3].

В статье рассматривается программа [4], предназначенная для расшифровки и оперативного анализа выходной информации РЛС, зарегистрированной КЗА. Программа выполняет считывание информации, полученной от КЗА РЛС, разбор записей подсистемы регистрации, т.е. приведение их к виду, удобному для анализа. Разбор заключается в выделении и проверке блоков данных, анализе состава зарегистрированной информации, преобразовании, сортировке по типам сообщений, отбраковке некорректной, поврежденной информации [5].

1. Описание зарегистрированной информации

Исходной информацией для анализа являются потоки данных и служебной информации, типовые структуры которых приведены на рис. 1.

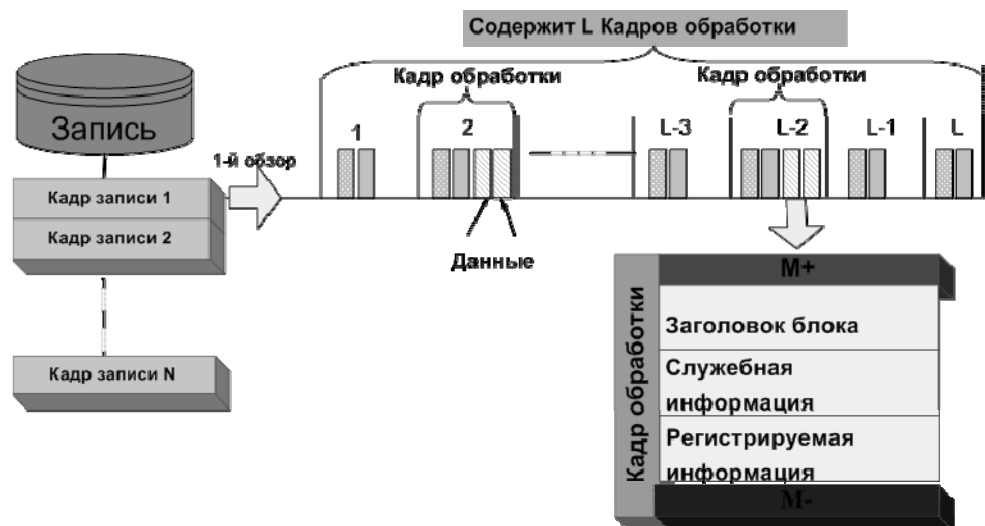


Рис. 1. Структура блока исходных данных

Первичная информация представляет собой совокупность данных, зарегистрированных за один сеанс-запись. Записи представляют собой файлы размером до 35 МБ. Внутренняя структура этих файлов различается в зависимости от режима работы РЛС. Запись состоит из информационных блоков, соответствующих каждому обороту антенны, – кадров записи. Кадр записи представляет собой отдельный файл, в котором записаны кадры обработки, разделенные маркерами. Для каждой структурной единицы имеется свой заголовок, описывающий размер блока и служебную информацию РЛС. Предлагается ускорить процедуру разбора радиолокационных записей за счет применения параллельной обработки.

2. Применение графических процессоров для обработки информации

В настоящее время активно применяются многопоточные модели для увеличения производительности систем [6]. В подавляющем большинстве случаев для организации параллельной обработки данных и вычислений используются многоядерные мультипроцессорные системы, основанные на центральных процессорах (CPU). С каждым годом производители CPU увеличивают их производительность за счет увеличения тактовой частоты, количества ядер, кэша и других технических характеристик. Нельзя исключать и вклад применения современных технологий, оптимизирующих работу процессоров. Однако в последние годы производители вынуждены искать замену традиционным средствам повышения быстродействия. Ограничения на потребляемую мощность и на тепловыделение, а также быстро приближающийся физический предел размера транзистора заставляют научных деятелей и производителей искать решение в другом месте [7].

Одним из возможных решений в данной проблеме может служить использование графических процессоров (GPU). Производители GPU пошли по пути параллелизма гораздо раньше. В сегодняшних видеокартах число процессоров может достигать 2304 (GeForce GTX 780). Производительность подобных систем при умелом их программировании может быть весьма значительной. Два крупнейших производителя GPU – NVIDIA и AMD – разработали программно-аппаратные архитектуры параллельных вычислений (CUDA от NVIDIA и FireStream от AMD), которые позволяют существенно увеличить вычислительную производительность благодаря использованию GPU. STREAM из-за отставания от CUDA в прошлом в настоящее время практически не используется.

Технология CUDA эффективно применяется для быстрого исполнения большого числа параллельно выполняемых потоков инструкций над различными данными. Таким образом, CUDA идеально подходит для решения проблемы дешифрации и анализа радиолокационных данных. При анализе и дешифрации информации, полученной от контрольно-записывающей аппаратуры бортовой РЛС, возникает проблема, которая состоит в обработке большого объема информации ввиду высокой скорости поступления данных в бортовую информационно-управляющую систему [8].

В первую очередь для решения поставленной проблемы с использованием технологии CUDA необходимо провести анализ данных и выделить независимые блоки.

Количество нитей в блоке и количество блоков в сетке выбрано исходя из максимизации загрузки ресурсов GPU и с учетом аппаратных ограниче-

ний. В связи с тем, что количество кадров обработки не превышает максимально допустимого количества нитей GPU, каждый кадр обработки обрабатывает отдельная нить. Кадры обработки распределены по нитям, так как в функции их анализа и дешифрации практически отсутствуют ветвления, что позволяет избавиться от «холостых» проходов нитей внутри варпов (warp). Распределение исходных данных между нитями и блоками представлено на рис. 2.

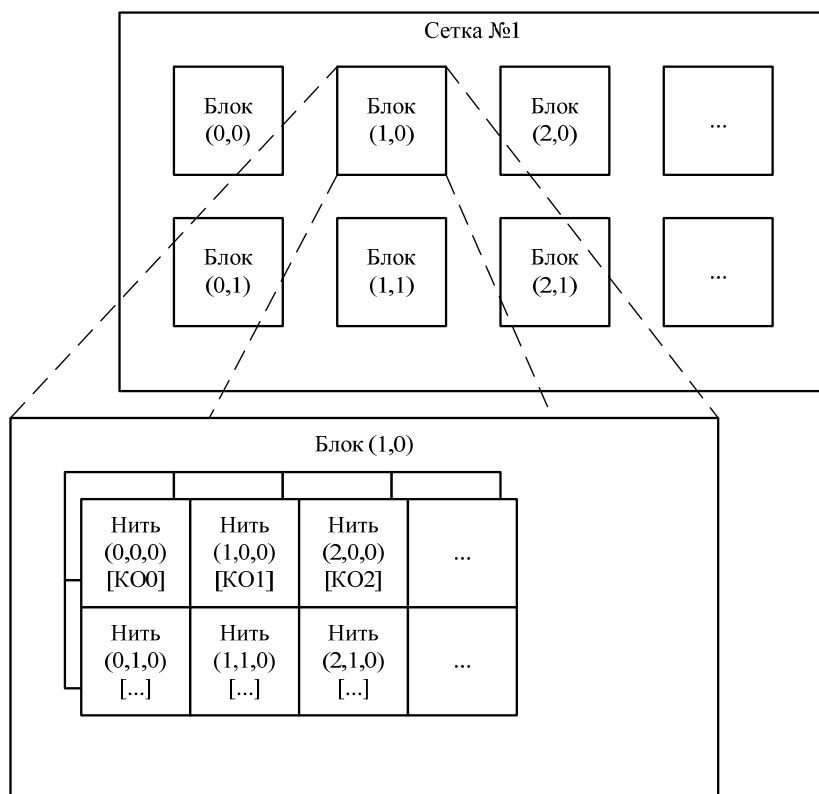


Рис. 2. Распределение данных между нитями и блоками (КО – кадр обработки)

Программное обеспечение анализа зарегистрированной радиолокационной информации реализовано на языке программирования C# в среде SharpDeveloper и обеспечивает эффективное использование вычислительных возможностей графических процессоров CUDA [4].

Программы для CUDA написаны на «расширенном» языке C и компилируются при помощи собственного Си-компилятора командной строки *nvcc*, входящего в состав CUDA SDK, файлы имеют расширение *.cu.

Для работы с графическим процессором сначала необходимо создать объект класса CUDA для нулевого видеоадаптера:

```
CUDA cu = new CUDA( 0, true );
```

Копируем входной массив данных в память видеоадаптера:

```
CUdeviceptr kz = cu.CopyHostToDevice (mas);
```

После этого необходимо загрузить подготовленную на стадии компиляции сборку *razbor.cubin*:

```
cu.LoadModule( "razbor.cubin" );
```

Поиск маркеров осуществляется на GPU в функции *SearchMarker*. Для вызова этой функции предварительно требуется получить ее дескриптор вызовом метода *GetModuleFunction*:

```
CUfunction srch = cu.GetModuleFunction("SearchMarker");
```

после чего необходимо указать параметры запуска:

```
cu.SetParameter( srch, 0, ( uint )kz.Pointer );
```

```
cu.SetFunctionBlockShape(srch, pr.MaxThreadsPerBlock, 1, 1 );
```

```
cu.SetParameterSize(srch, ( uint )( IntPtr.Size + sizeof( uint ) ) );
```

```
cu.SetParameter(srch, IntPtr.Size, ( uint )42 );
```

Для начала необходимо определить размер блока – это будет совокупность из максимального количества нитей. Таким образом, обработка входной последовательности данных будет производиться блоками по *pr.MaxThreadsPerBlock* элементов. После чего происходит запуск функции на исполнение и результат копируется обратно в память хоста:

```
cu.Launch( func, mas.Length);
```

```
cu.CopyDeviceToHost (kz, mas );
```

Данный вызов метода является синхронным, поэтому возврат из метода не будет произведен, пока ядро не обработает все элементы массива. Затем происходит освобождение ранее выделенной памяти (метод *Free*):

```
cu.Free( kz );
```

3. Алгоритм анализа и дешифрации данных на GPU

Предлагается следующий алгоритм анализа и дешифрации данных на GPU:

1. Загрузка в глобальную память GPU-файла, полученного от контрольно-записывающей аппаратуры РЛС.

2. Поиск маркеров.

Маркер имеет размер 8 байт и представляет собой определенную последовательность бит. Перед анализом и дешифрацией предварительно необходимо найти все маркеры начала кадров обработки. Поиск маркеров также возложен на CUDA. Исходные данные разбиваются на максимальное количество частей.

Максимальное количество нитей GPU определяется следующим образом: *pr.MaxThreadsPerBlock*.

Каждую часть обрабатывает одна нить в одном из блоков, считывая и анализируя по 8 байт данных за один проход. В случае совпадения 8 байт со значением маркера сохраняется позиция начала кадра обработки. На практике для нахождения всех маркеров требуется всего один проход. По завершении

поиска маркеров запускается непосредственно анализ, преобразование и дешифрация данных:

```
extern "C" __global__ void SearchMarker(int ln)
{ // Маркеры начала
  char start[] = {0xE5, 0xC5, 0x49, 0xBB, 0x59, 0xFD, 0x20, 0x8C};
  // Значение маркера {0x8C, 0x20, 0xFD, 0x59, 0xBB, 0x49, 0xC5, 0xE5};
  int ind = blockIdx.x * blockDim.x + threadIdx.x;
  int ind2;
  int k = ind;
  if (ind > ln) return;
  for (int i = 0; i < sizeof(start); i++){
    if (start[i] != mas[k]) {
      return;
    } k++;
  } // Начало разбора данных
  ...
} // Запись результатов
```

3. Дешифрация и анализ данных.

При разборе кадра обработки выполняется последовательное чтение полей данных согласно протоколу КЗА РЛС. Данные проверяются на корректность и записываются в соответствующие поля выходных структур. В случае повреждения информации или ее некорректности весь кадр обработки бракуется и целиком записывается в отдельный буфер.

4. Формирование выходных файлов по результатам анализа и дешифрации.

По завершении анализа и дешифрации кадра обработки выполняется копирование выходных структур из памяти GPU в оперативное запоминающее устройство (ОЗУ) CPU с последующей записью в результирующие файлы. Иерархия каталогов разобранного сеанса записи представлена в табл. 1.

4. Оценка времени обработки на GPU

Для оценки времени обработки данных на GPU необходимо создать объекты класса *cudaEvent_t* и произвести отсечки времени до и после обработки данных, используя соответствующие функции запуска и остановки времени расчетов:


```
// События для синхронизации и замера времени работы GPU
cudaEvent_t evStart;
cudaEvent_t evStop;
cudaEventCreate(&evStart);
```

```

cudaEventCreate(&evStop);
// старт расчетов на GPU
cudaEventRecord(evStart, 0);
// Остановка времени расчетов на GPU
cudaEventRecord(evStop, 0);
    
```

Таблица 1

Иерархия каталогов разобранного сеанса записи

Каталог	Примечание
 zz000	Каталог записи (000 – номер сеанса записи)
 kz000	Каталог кадра записи (000 – номер кадра записи)
 ko000	Каталог кадра обработки (000 – номер кадра обработки)
 fo.dat	Файл формуляра кадра обработки (Формуляр цели)
 zo.dat	Файл заголовка кадра обработки
 k1.dat	Файл данных кадра обработки первого канала
 k2.dat	Файл данных кадра обработки второго канала
 k3.dat	Файл данных кадра обработки третьего канала
 k4.dat	Файл данных кадра обработки четвертого канала
 ko001	Каталог кадра обработки
 ko002	Каталог кадра обработки
 ko003	Каталог кадра обработки
***	***
 ko00N	Каталог кадра обработки
 zfo.dat	Файл таблицы формуляров
 kz001	Каталог кадра записи
 kz002	Каталог кадра записи
***	***
 kz00N	Каталог кадра записи
 zz.dat	Файл заголовка записи
 log.txt	Лог файл
 bad.dat	Файл поврежденной информации

Работоспособность алгоритма протестирована на тестовых наборах входной информации от 10 до 110 кадров обработки. Программа выполнялась на графическом процессоре на видеокарте GeForce GTX 480 (480 ядер) и на центральном процессоре Intel Core i3-4330. По результатам тестирования был построен график (рис. 3). По графику видно, что чем больше количество кадров обработки, тем выполнение программы на GPU эффективнее CPU. Так как время, затрачиваемое на обработку данных в GPU, не увеличивается с числом кадров обработки в связи с тем, что каждый кадр обработки анализирует отдельная нить в блоке CUDA. В реальных полетных данных в одном кадре записи не менее 128 кадров обработки, соответственно использование

GPU для параллельной обработки данных дает прирост производительности до 10 % (рис. 4). Основное время работы на GPU тратится на операции копирования входного набора данных в память видеоадаптера, прирост производительности с увеличением кадров обработки не будет больше 10 %.

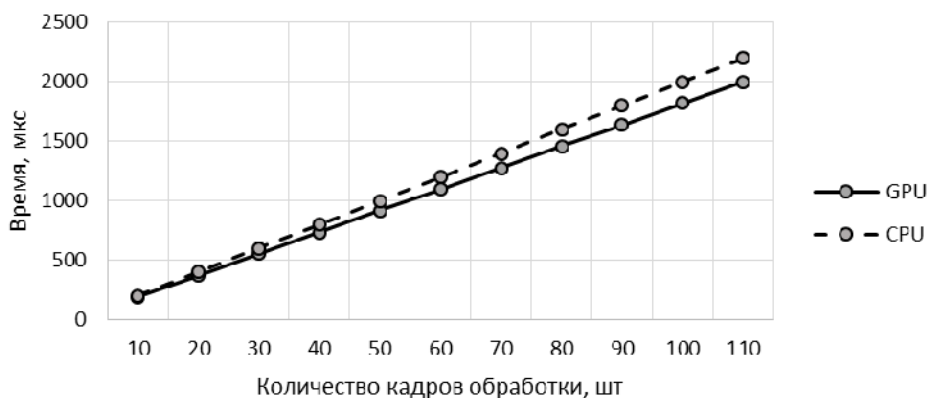


Рис. 3. Зависимость времени выполнения программы от количества кадров обработки

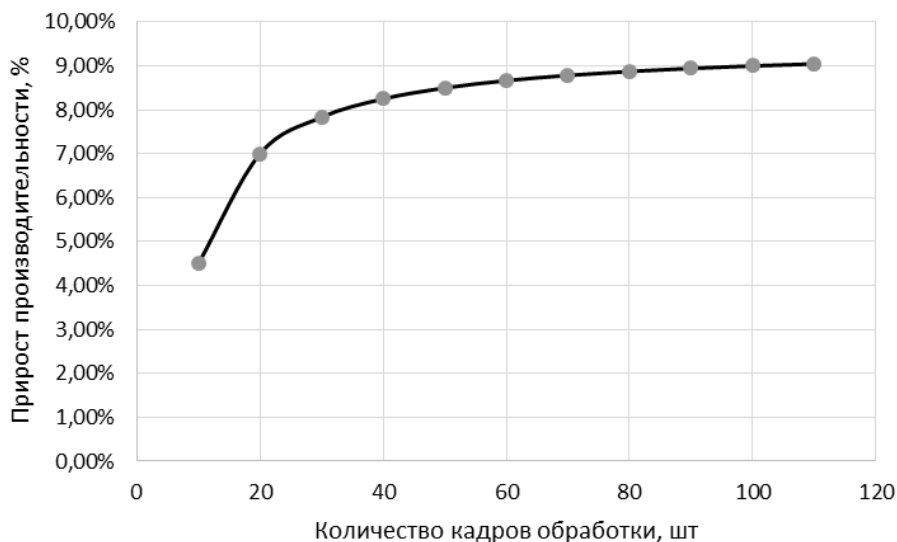


Рис. 4. Прирост производительности GPU относительно CPU

На рис. 3 приведено сравнение производительности алгоритма дешифрации и анализа данных РЛС на центральном процессоре и на графическом.

Заключение

Сравнивая различные подходы к распараллеливанию процесса дешифрации и анализа радиолокационных данных на CPU [9] и GPU, обобщим их достоинства и недостатки. Основное преимущество в использовании GPU перед CPU заключается в высокой степени распараллеливания. Одновременно может выполняться анализ нескольких десятков тысяч кадров обработки. Другим преимуществом является относительная простота реализации алго-

ритма с использованием CUDA-технологии, а также отсутствие применения дополнительного дорогостоящего оборудования (используются ресурсы имеющегося видеоадаптера).

Недостатком GPU является относительно медленный доступ к глобальной памяти. Тем не менее, несмотря на затрачиваемое время на копирование блоков памяти в ОЗУ GPU, применение технологии CUDA оказалось более эффективным, чем использование параллельной обработки данных на CPU [10].

Список литературы

1. **Верба, В. С.** Авиационные комплексы радиолокационного дозора и наведения / В. С. Верба. – М. : Радиотехника, 2008. – 432 с.
2. **Чернышев, М. И.** Аппаратно-программный комплекс регистрации и имитации радиолокационных сигналов / М. И. Чернышев, Л. Н. Кадников, Н. Н. Коннов // Радиолокация, навигация и связь (RLNC-2006) : сб. докладов XII Междунар. науч.-техн. конф. – Воронеж, 2006. – Т. 2. – С. 1169–1173.
3. **Коннов, Н. Н.** Программный комплекс для анализа сигналов РЛС / Н. Н. Коннов, Р. Н. Федюнин, Е. К. Таранцев, М. И. Чернышев // Новые информационные технологии и системы : тр. IX Междунар. науч.-техн. конф. – Пенза, 2010. – Ч. 2. – С. 195–200.
4. Программа дешифрации радиолокационной информации регистрируемой бортовой контрольно-записывающей аппаратурой : свид. о регистрации программы для ЭВМ № 2015611225. Зарег. 26 января 2015 г. / Е. П. Лепилина, Н. Н. Коннов, К. В. Попов, В. Б. Механов. – М. : Роспатент, 2015.
5. **Фирстова, Е. П.** Программа для расшифровки данных контроля радиолокационного комплекса / Е. П. Фирстова, Н. Н. Коннов // Новые информационные технологии и системы : сб. докладов IX Междунар. науч.-техн. конф. – Пенза, 2010. – С. 200–204.
6. **Лепилина, Е. П.** Проблемы построения многопоточной модели программы дешифрации радиолокационных данных / Е. П. Лепилина // Современные проблемы компьютерных наук (СПКН-2013) : сб. материалов I Междунар. науч.-практ. конф. – Пенза : Изд-во ПГУ, 2013. – С. 116–119.
7. **Сандерс, Дж.** Технология CUDA в примерах. Введение в программирование графических процессоров / Дж. Сандерс, Э. Кэндрот. – М. : ДМК Пресс, 2013. – 232 с.
8. **Таранцев, Е. К.** Способы повышения производительности программно-аппаратных комплексов РЛС импульсно-доплеровского типа / Е. К. Таранцев, Н. Н. Коннов // Телекоммуникации. – 2011. – № 5. – С. 25–33.
9. **Лепилина, Е. П.** Проблема повышения производительности системы дешифрации и анализа радиолокационных данных / Е. П. Лепилина // Сборник материалов XI Международной научно-технической конференции, 17–20 сентября 2013 / Центр информационных технологий в проектировании РАН, Юго-Западный гос. ун-т. – Курск : ЮЗГУ, 2013. – 448 с.
10. Функциональное программное обеспечение подсистемы анализа радиотехнических сигналов РДПИ.01092-02 : свид. о регистрации программы для ЭВМ № 2011614499. Зарег. 13 апреля 2011 г. / Е. К. Таранцев, Н. Н. Коннов, М. В. Логунов, К. В. Попов, Р. Н. Федюнин, Р. И. Голутвин, Е. П. Фирстова, Т. В. Ембулаева. – М. : Роспатент, 2011.

References

1. Verba V. S. *Aviatsionnye komplekсы radiolokatsionnogo dozora i navedeniya* [Aircraft complexes of radar picket and guidance]. Moscow: Radiotekhnika, 2008, 432 p.

2. Chernyshev M. I., Kadnikov L. N., Konnov N. N. *Radiolokatsiya, navigatsiya i svyaz' (RLNC-2006): sb. докладov XII Mezhdunar. nauch.-tekhn. konf.* [Radar, navigation and communication (RNC-2006): proceedings of XII International scientific and technical conference]. Voronezh, 2006, vol. 2, pp. 1169–1173.
3. Konnov N. N., Fedyunin R. N., Tarantsev E. K., Chernyshev M. I. *Novye informatsionnye tekhnologii i sistemy: tr. IX Mezhdunar. nauch.-tekhn. konf.* [New information technologies and systems: proceedings of IX International scientific and technical conference]. Penza, 2010, part 2, pp. 195–200.
4. Lepilina E. P., Konnov N. N., Popov K. V., Mekhanov V. B. *Programma deshifratsii radiolokatsionnoy informatsii registriruemoy bortovoy kontrol'no-zapisyvayushchey apparatury: svid. o registratsii programmy dlya EVM № 2015611225. Zareg. 26 yanvaryaya 2015 g.* [Program of radar information decryption registered by aircraft recorder equipment: computer program registration certificate № 2015611225, registered on 26th January 2015]. Moscow: Rospatent, 2015.
5. Firstova E. P., Konnov N. N. *Novye informatsionnye tekhnologii i sistemy: sb. докладov IX Mezhdunar. nauch.-tekhn. konf.* [New information technologies and systems: proceedings of IX International scientific and technical conference]. Penza, 2010, pp. 200–204.
6. Lepilina E. P. *Sovremennye problemy komp'yuternykh nauk (SPKN-2013): sb. materialov I Mezhdunar. nauch.-prakt. konf.* [Modern problems of computer sciences (MPCS-2013): proceedings of I International scientific and practical conference]. Penza: Izd-vo PGU, 2013, pp. 116–119.
7. Sanders G., Kendrot E. *Tekhnologiya CUDA v primerakh. Vvedenie v programmirovaniye graficheskikh protsessorov* [The CUDA technology in examples. Introduction into graphic processor programming]. Moscow: DMK Press, 2013, 232 p.
8. Tarantsev E. K., Konnov N. N. *Telekommunikatsii* [Telecommunications]. 2011, no. 5, pp. 25–33.
9. Lepilina E. P. *Sbornik materialov XI Mezhdunarodnoy nauchno-tekhnicheskoy konferentsii, 17–20 sentyabrya 2013* [Proceedings of XI International scientific and technical conference, 17–20 September 2013]. Tsentr informatsionnykh tekhnologiy v proektirovaniy RAN, Yugo-Zapadnyy gos. un-t. Kursk: YuZGU, 2013, 448 p.
10. Tarantsev E. K., Konnov N. N., Logunov M. V., Popov K. V., Fedyunin R. N., Golutvin R. I., Firstova E. P., Embulaeva T. V. *Funktional'noe programnoye obespechenie podsistemy analiza radiotekhnicheskikh signalov RDPI.01092-02: svid. o registratsii programmy dlya EVM № 2011614499. Zareg. 13 aprelyaya 2011 g.* [Functional software for radar signal analysis sub-systems RDPI.01092-02: computer program registration certificate № 2011614499. Registered on 13th April 2011]. Moscow: Rospatent, 2011.

Лепилина Екатерина Павловна
аспирант, Пензенский
государственный университет (Россия,
г. Пенза, ул. Красная, 40)

E-mail: vkatret@yahoo.com

Lepilina Ekaterina Pavlovna
Postgraduate student, Penza State
University (40 Krasnaya street,
Penza, Russia)

УДК 004.42

Лепилина, Е. П.

Программа дешифрации и анализа радиолокационной информации с применением технологии CUDA / Е. П. Лепилина // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2015. – № 3 (35). – С. 15–24.