

# МДОЗМФ-2009

Подход к моделированию и  
реализации параллельных  
вычислений на языке Ada

С. И. Киркоров, Л. С. Киркорова

*Научно-производственное предприятие  
МедиаСкан, г. Минск, РБ.*

# Выбор инструментария

Одной из причин моделирования алгоритмов, реализующих параллельные вычисления, является необходимость их дальнейшего применения в аппаратно-программных или аппаратных системах.

Особенность аппаратных средств – это возможность параллельного функционирования двух и более модулей системы, причем любой модуль может сам состоять из параллельно работающих частей.

Одной из причин ограничений применения стандартного инструментария для моделирования алгоритмов это сложность или высокая стоимость обеспечения соответствующего уровня гарантий конечного продукта в области технической защиты информации по всем трем составляющим – конфиденциальность, целостность и доступность.

# Выбор инструментария

Для тестирования доверенной платформы и параллельных вычислений реализуемые ее, как правило, создается стендовое оборудование. Стендовое оборудование как средство измерения также требует соответствующего уровня гарантий правильности своего функционирования.

Правильным выбором в этом случае было бы использовать инструментальное средство, которое:

1. Само могло бы пройти сертификационные испытания в области технической защиты информации, то есть, как минимум имела открытые спецификации, коды программ и так далее.
2. Соответствовало стандарту, строго его выполняло и имело стандарт, регламентирующий эти проверки. Было реализовано (или могло быть адаптировано) для целевых платформ.
3. Обеспечивало поддержку многозадачности (для моделирования алгоритмов, реализующих параллельные вычисления) на уровне языка высокого уровня. Этим достигается стабильность семантики, разработчик избавляется от необходимости использования разнородных внешних библиотек или собственных решения для обеспечения многозадачности.
4. Влияет, конечно, и экономический фактор. Стоимость приобретения и совокупного владения такого инструментария не должна превышать  $1/3$  от всей стоимости разработки.

Встроенная поддержка многозадачности является уникальной и широко известной особенностью языка программирования Ada, которая выгодно отличает его от большинства современных языков программирования.

# Язык Ada в учебном процессе

## Исследования по улучшению локальности многомерных алгоритмов с помощью тайлинга.

Под тайлом понимается множество операций алгоритма, выполняемых как одна единица вычислений атомарно. Геометрически тайл можно представить множеством точек многомерной целочисленной решетки. Наиболее частой для использования является форма прямоугольного параллелепипеда.

Исходными данными к работе послужили способы улучшения локальности многомерных циклов: перестановка циклов, тайлинг.

Лиходед Н.А. Методы распараллеливания гнезд циклов: Курс лекций. – Мн.: БГУ. 2007. – 100 с. Ser314\ subFaculty\ Каф. Дискр. мат. и алгор\ КУРСЫ ДМА\ 4 курс\ Лиходед\ Лекции\ Распараллеливание гнезд циклов

Ahmed N., Mateev N., Pingali K. Synthesizing transformations for locality enhancement of imperfectly-nested loop nests. Proceedings of the International Conference on Supercomputing. 2000. P. 141–152.

# Язык Ada в учебном процессе

В результате исследования подтвердилось влияние перестановки циклов и тайлинга на скорость работы алгоритма. Оптимальные решения были разные в зависимости от аппаратной платформы (в данном исследовании сравнение проводилось на одноядерном и двухъядерном процессорах фирмы Intel). Проведенные экспериментальные исследования позволили определить максимальные значения количества тайлингов выполняемых в виде задачи Ada-95 и их оптимум для конкретных сочетаний программных и аппаратных средств. Ниже приведен текст тестового модуля (сокращенный) для оценки производительности аппаратно-программной платформы:

```
subtype tiling_count is Integer range 0..12;  
type tiling_type is array (tiling_count) of integer;  
tiling : tiling_type :=  
    (2,4,5,8,10,20,25,40,100,125,200,250,500);
```

# Язык Ada в учебном процессе

(продолжение)

```
for ct in 5..12 loop
  CalendD1 := Ada.Calendar.Clock; -- начальное время
  Test.matr_multipl_TASK(a,b,c,Test.count,tiling(ct));
  CalendD2 := Ada.Calendar.Clock; -- конечное время
  time_taling := Test.Time_sec(CalendD1,CalendD2);
end loop;
```

```
procedure matr_multipl_TASK
(a : access taling_array;
 b : access taling_array;
 c : access taling_array;
 count : Integer;
 tile : Integer
) is
  tilsize : Integer := count/tile;
  i1begin, i1end : integer := 0;
  j1begin, j1end, k1begin, k1end : integer := 0;
  task type multipl_tiling (ind1, ind2, ind3 : integer) ;
```

# Язык Ada в учебном процессе

(продолжение)

```
type multipl_tiling_Ptr is access multipl_tiling;
type type_ar_tsk is array
  (1..tilesize,1..tilesize,1..tilesize) of multipl_tiling_Ptr;
ar_tsk : type_ar_tsk;
begin
  for i1 in 0 .. tilesize-1 loop
    i1begin := i1*tile;
    i1end := (i1+1)*tile - 1;
    for j1 in 0 .. tilesize-1 loop
      j1begin := j1*tile;
      j1end := (j1+1)*tile - 1;
      for k1 in 0..tilesize-1 loop
        k1begin := k1*tile;
        k1end := (k1+1)*tile - 1;
        ar_tsk(i1+1,j1+1,k1+1) := new multipl_tiling(i1,j1,k1);
      end loop;
    end loop;
  end loop;
end matr_multipl_TASK;
```

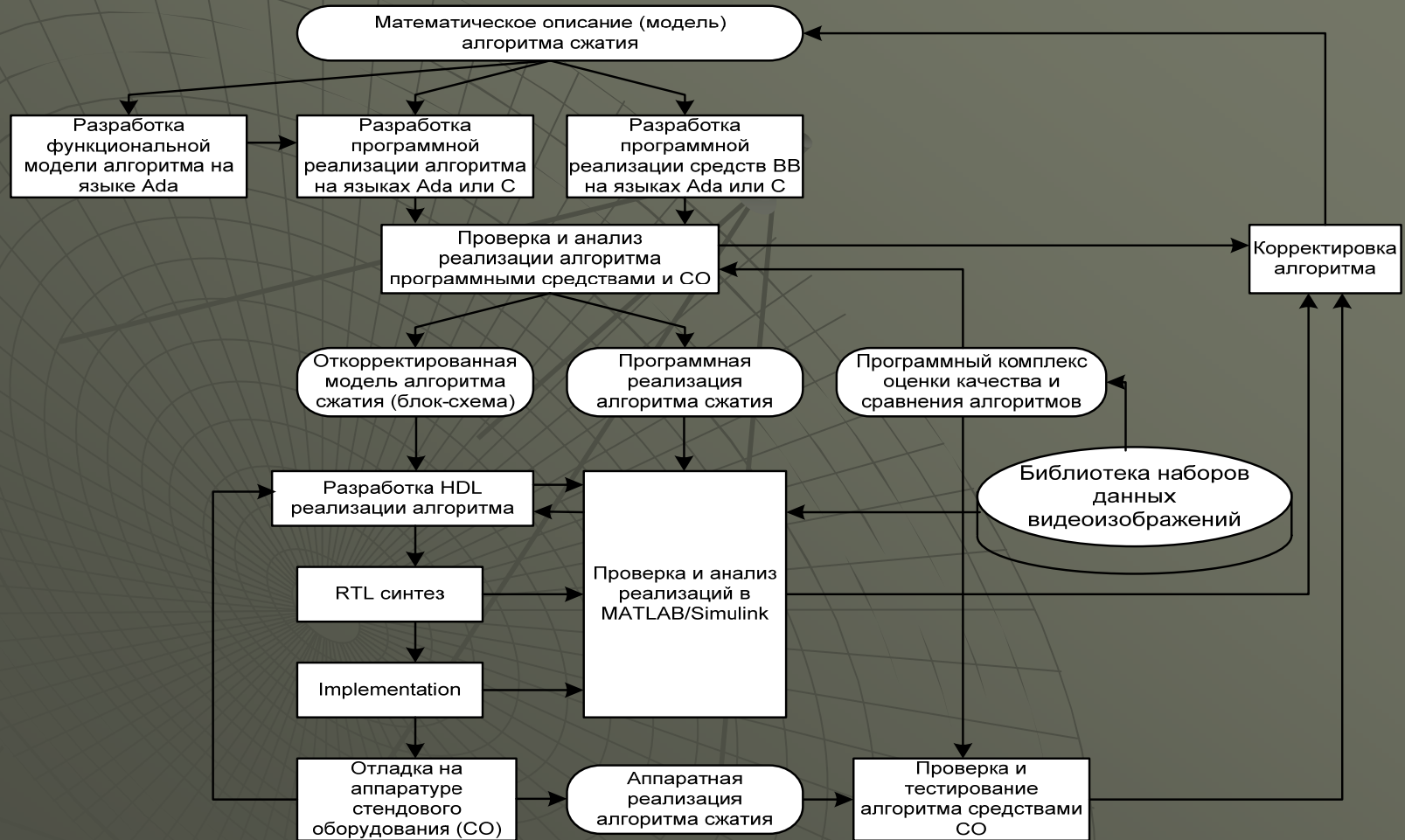
# Язык Ada в НИОКР

Использование тайлинга для улучшения локализации алгоритма значительно повышает скорость расчетов. При разной последовательности циклов в алгоритме оптимальный размер тайла не одинаков для всех случаев. В каждом случае он разный. Поэтому для каждой модификации алгоритма при использовании тайлинга необходимо экспериментально подбирать его размер. И тогда время вычислений будет улучшено.



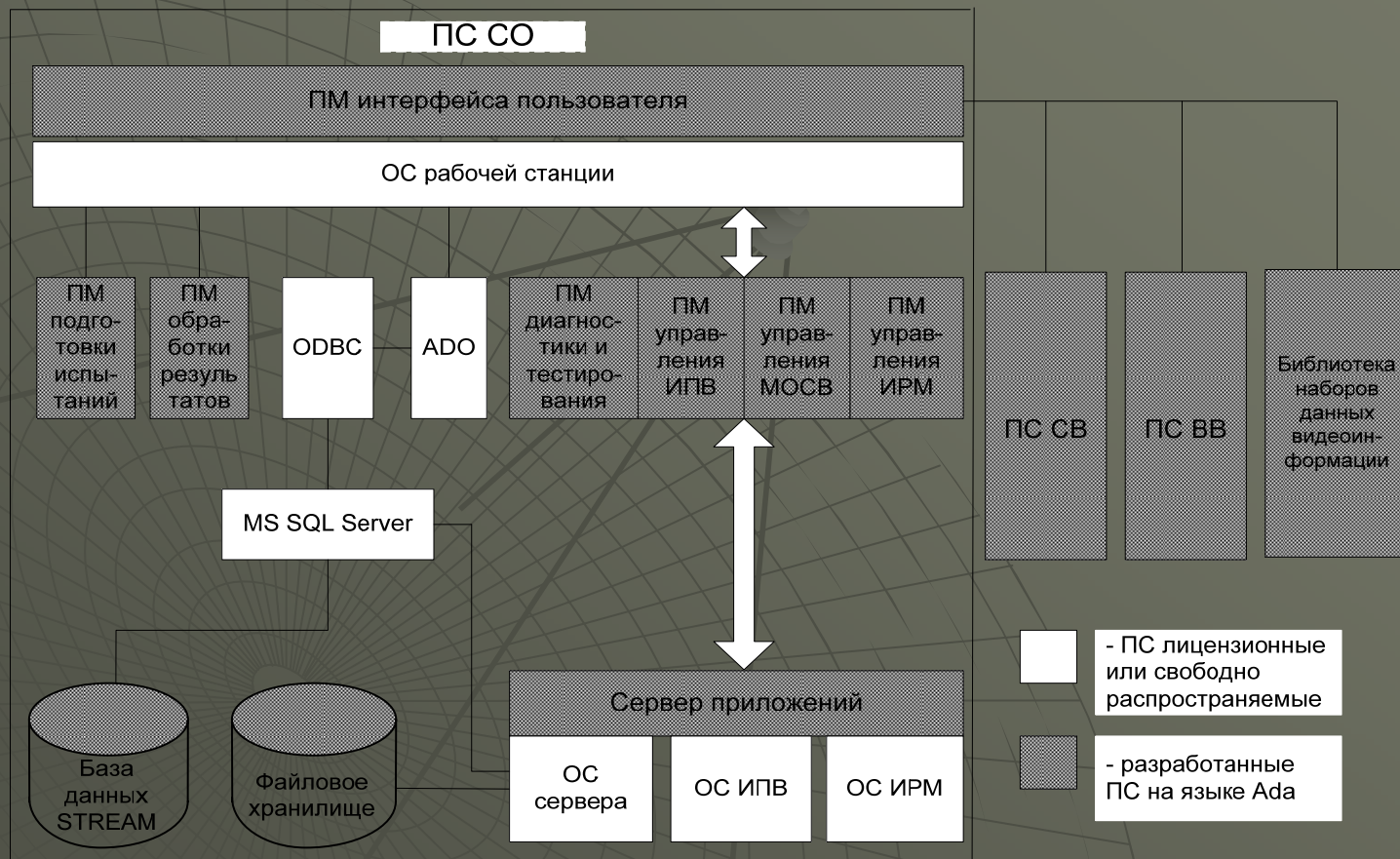
# Язык Ada в НИОКР

## Процесс разработки алгоритмов



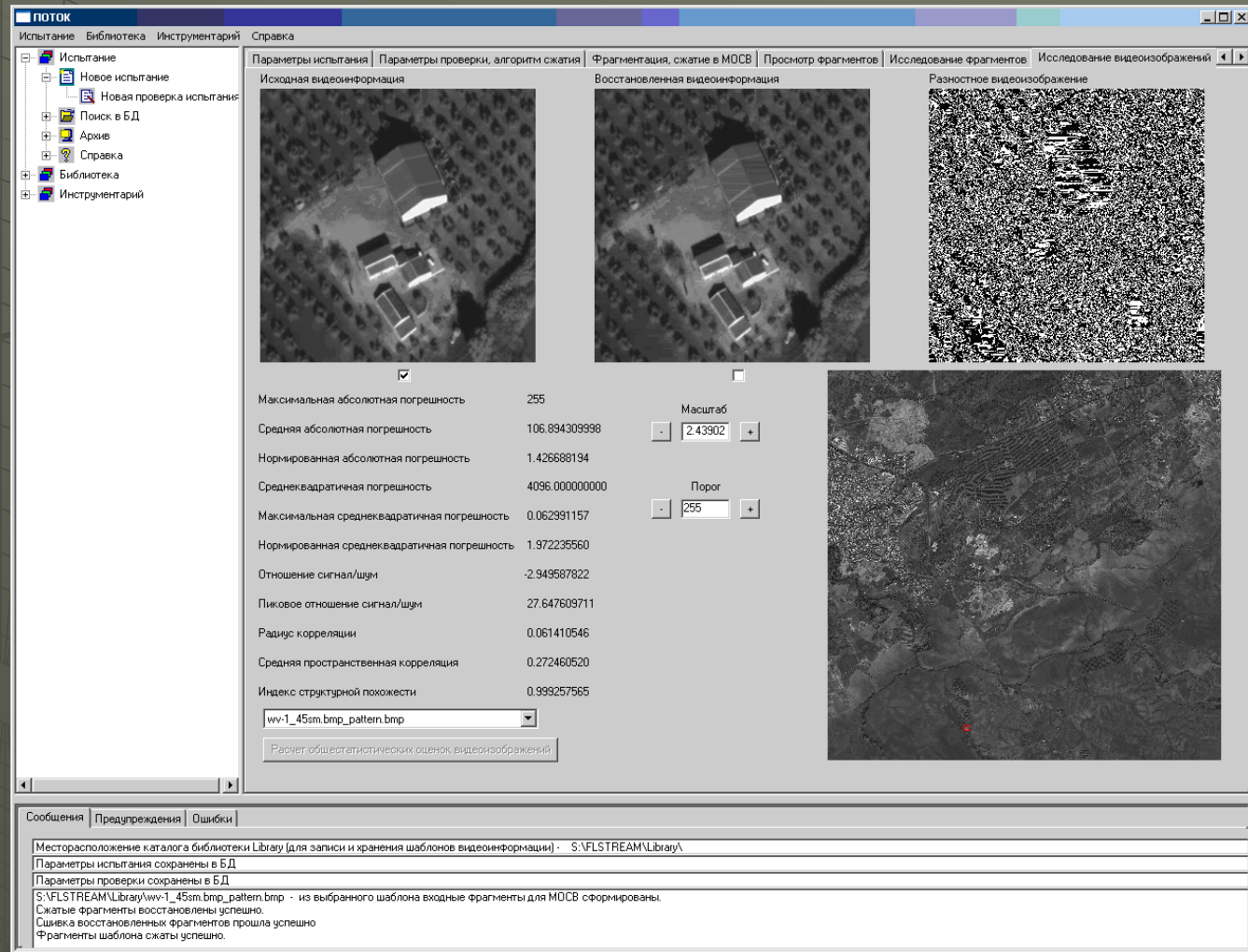
# Язык Ada в НИОКР

Структура программных средств стандового оборудования .



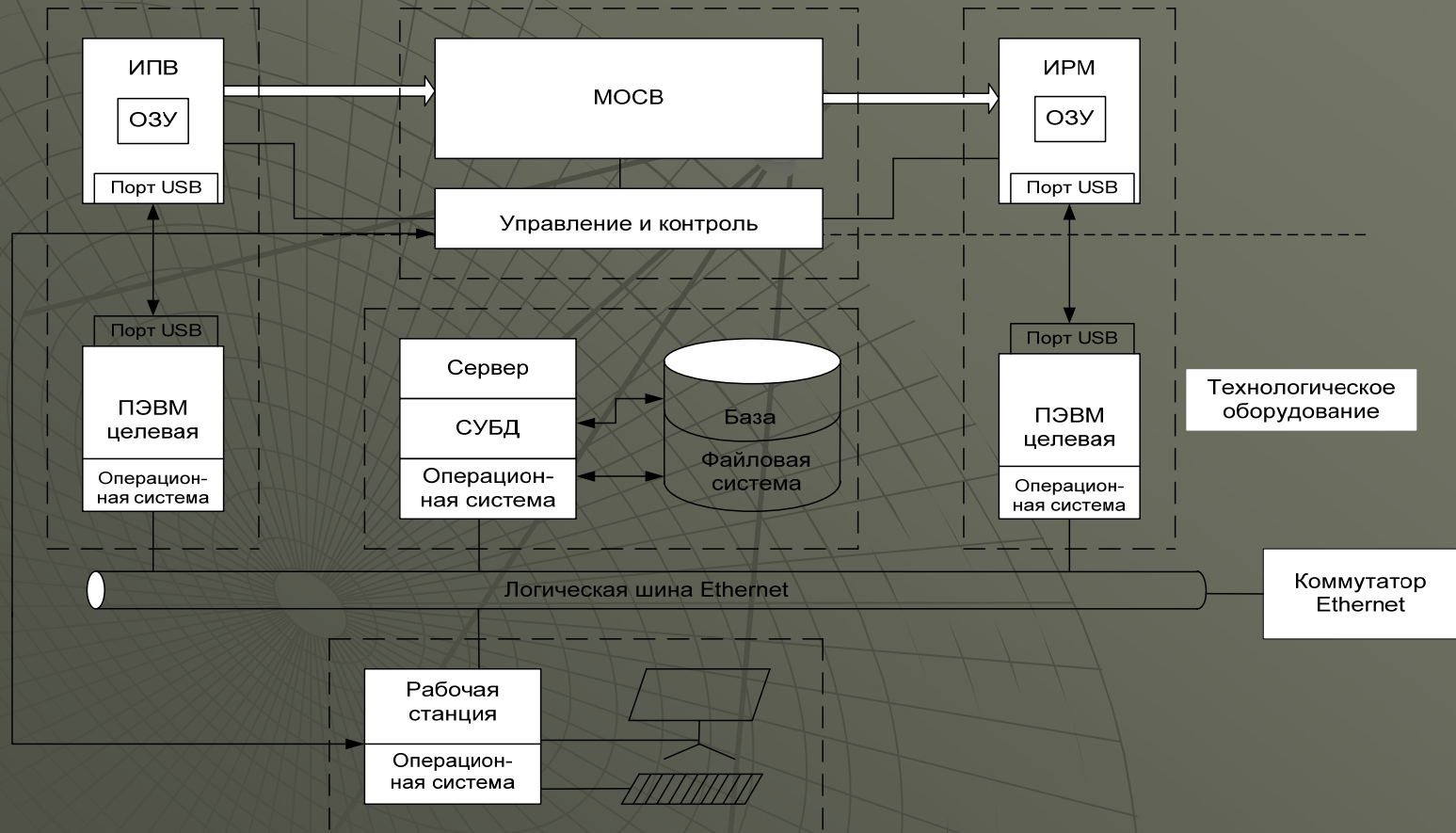
# Язык Ada в НИОКР

- Интерфейс пользователя разработан на языке Ada с использованием пакета GWindows



# Язык Ada в НИОКР

Структура аппаратных средств стендового оборудования .



# Язык Ada в НИОКР

Внешний вид стендового оборудования

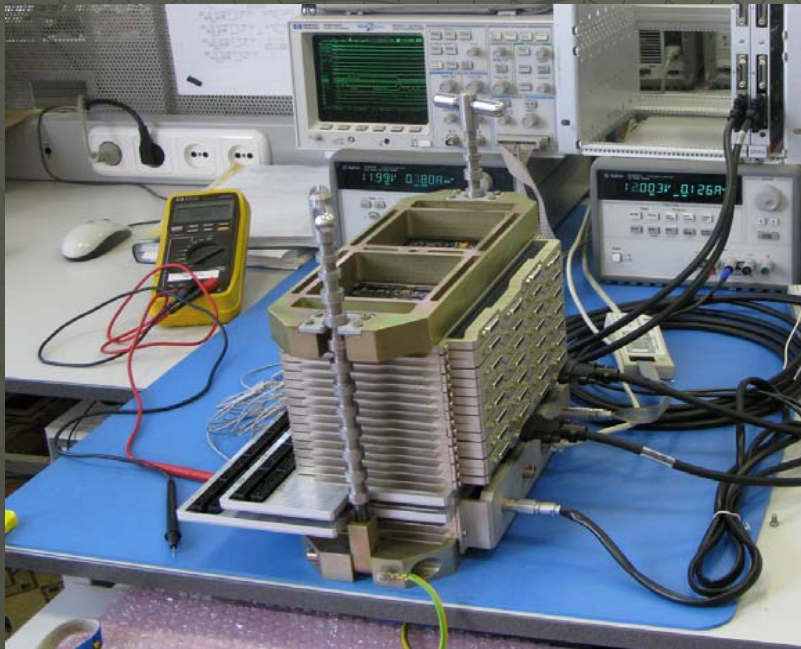


# Язык Ada в НИОКР

Таким образом, использовалась программная модель, реализующая основные вычисления на языке Ada. Отработанная программная модель адаптировалась под конкретный кристалл FPGA на языке VHDL.

Обмен информации в/из МОСВ выполнен в формате данных протокола из рекомендаций CCSDS-133.0-B-1.

Внешний вид МОСВ (макетный образец сжатия видеoinформации) в процессе прошивки алгоритмов и в собранном виде.



# МДОЗМФ-2009

## ЛИТЕРАТУРА

1. Элементы параллельного программирования/ В.А. Вальковский, В.Е. Котов, А.Г. Марчук, Н.Н. Миренков; Под ред. В.Е. Котова. – М.: Радио и связь, 1983. – 240 с., ил.
2. СТБ 34.101.3, стандарт РБ.
3. ИСО/МЭК 15408-std.
4. Department of Defense Trusted Computer Security Evaluation Criteria. (Orange Book). – 12/85 (Dog 5200.28-std) – Rainbow Series.
5. Ada 95 Language Reference Manual ANSI/ISO 8652.1995-std.  
<http://www.adapower.com/rm95/index.html>.
6. Гавва А. Е. “Адское” программирование. Ada-95. Компилятор GNAT. <http://www.ada-ru.org>.
7. Киркорова Л. С. Экспериментальные исследования локальности алгоритмов перемножения матриц, реализованных на двухъядерных компьютерах. Курсовая работа. – Мн.: БГУ. 2008.
8. Лиходед Н.А. Методы распараллеливания гнезд циклов: Курс лекций. – Мн.: БГУ. 2007. – 100 с.  
Ser314\ subFaculty\ Каф. Дискр. мат. и алгор\ КУРСЫ ДМА\ 4 курс\ Лиходед\ Лекции\  
Распараллеливание гнезд циклов
9. Ahmed N., Mateev N., Pingali K. Synthesizing transformations for locality enhancement of imperfectly-nested loop nests. Proceedings of the International Conference on Supercomputing. 2000. P. 141–152.
10. Язык описания аппаратуры цифровых систем – VHDL. Описание языка (ГОСТ Р 50754-95), стандарт, М., 1995 г.
11. Gregory P. Chapelle, Michael L. Lewis. Software Metrics Lead the Way to Better HDL Coding Practices – Integrated System Design Magazine, 2000.
12. Robert J. Sheraga ANSI C to Behavioral VHDL Translator, Ada to Behavioral VHDL Translator. The RASSP Digest – Vol. 3. September 1996.

# МДОЗМФ-2009

Спасибо за внимание