

# Автоматическое отображение программ на процессор с ПЛИС-ускорителем

А.С. Рошаль, Д.В. Дубров, Р.Б. Штейнберг, Б.Я. Штейнберг  
Южный Федеральный Университет  
Ростов-на-Дону  
2014 г.

# Постановка задачи

## Проблема:

- Долгое время перепрограммирования ПЛИС
- Разработка высокопроизводительных ПЛИС
- Сложность проектирования ПЛИС

## • Решение:

- Использование специализированных приложений для автоматической HDL кодогенерации из программ написанных на я.п. более высокого уровня в я.п. более низкого уровня (C → VHDL)

# Описание системы

## Система кодогенерации

- Оснавана на диалоговом высокоуровневом оптимизирующем параллеливателе программ
- Включает:
  - ' - Front-ends(C and Фортран 2000)
  - Внутреннее представление программы Reprise
  - ' - Анализатор кода
    - ' - Анализ зависимостей
    - ' - Граф вычислений
    - ' - И др.
  - ' - Преобразование кода
  - ' - Back-ends(HDL генерация)

## Ограничения на входные данные

- Конвейеризуемый самый глубоко вложенный цикл
- - Целочисленная арифметика
- - Линерализированные индексы в выражениях  $x[a * i + b * j + k]$

# Алгоритм HDL-кодогенерации

- 7 шагов:
  - ' - Разбор входной программы на я.п. С
  - ' - Поиск возможного конвейеризируемого цикла
  - ' - Построение графа зависимостей для тела этого цикла
  - ' - Проверка тела цикла на возможность конвейеризируемости
  - ' - Получение графа вычислений тела цикла для вычисления стартовых задержек с учетом оптимизаций
  - ' - Получение Reprise представления для HDL кода
  - ' - Преобразование Reprise представления в текстовое представление VHDL

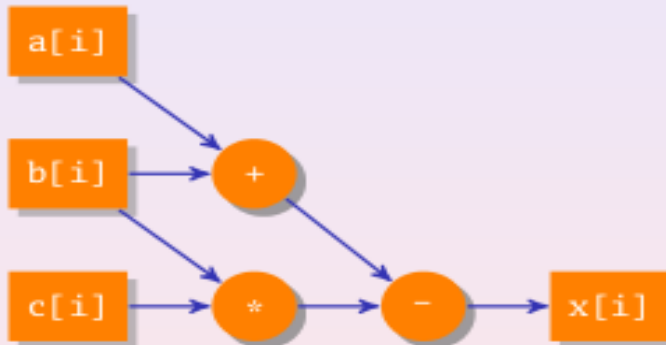
# Дополнительные оптимизации

- ┆ Интервальный анализ
  - ┆ Использует интервальную алгебру
  - ┆ Работает на этапе прекомпиляции
  - ┆ Расчитывает разряд переменных
- ┆ Расчет площади на кристалле
  - ┆ Позволяет сравнить площадь на кристалле двух фрагментов кода и выявить минимальный

# Пример

```
int main()
{
int i;
int a[1000], b[1000], c[1000], x[1000];
for (i = 0; i < 1000; i = i + 1)
x[i] = (a[i] + b[i]) - (c[i] * b[i]);
}
```

Программа на я.п. С



Граф вычислений

Operation	Delay
Read a[i]	0
Read b[i]	0
Read c[i]	0
+	1
*	1
-	6
Write x[i]	9

Зависимость между операциями и задержками

```
entity Sub1 is architecture Sub1_synth of Sub1 is
port( signal x_Out_Ready: std_logic;
CLK: in std_logic; signal x_Out_Ctr:
RST: in std_logic; unsigned(3 downto 0);
Start: in std_logic;
a: in signed(15 downto 0);
a_In_Ready: in std_logic;
x: out signed(15 downto 0);
Out_Ready: out std_logic);
end Sub1;
```

Выходной VHDL код

# Дальнейшее движение

- 1) Применение высокоуровневых преобразований кода для оптимизаций
- 2) Кодогенерация для различного оборудования в зависимости от критериев оптимизации
- 3) Реализация локальной и глобальной памяти
- 4) Расширение класса входных программ -  
новые типов данных
  - новые структур данных
  - условные операторы



# Литература

- Каляев А.В., Левин И.И. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений // М., «Янус-К», 2003, 380 с.
2. Штейнберг Р. Б. Отображение гнезд циклов на многоконвейерную архитектуру, Программирование, 2010, No 3. Steinberg R. Mapping loop nests to multipipelined architecture. // MAIK Nauka/Interperiodica distributed exclusively by Springer Science+Business Media LLC., May 2010, Vol 36, No3, pp 177-185.
3. Яджак М.С. Высокопараллельные алгоритмы и методы для решения задач массовых арифметических и логических вычислений // Диссертация на соискание ученой степени д.ф.-м.н. Институт прикладных проблем механики и математики. Львов. 2009г – 298с. (на украинском языке).
4. Bondalapati K. Modeling and Mapping for Dynamically Reconfigurable Hybrid Architecture. Ph.D.Thesis, University of Southern California, August, 2001.
5. Dubrov D.V., Roshal A.V., Generating Pipeline Integrated Circuits Using C2HDL Converter // Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2013), Rostov-on-Don, Russia, September 27-30, 2013, p. 215-219.
6. Self R. P., Fleury M., Downton A. C. A Design Methodology for Construction of Asynchronous Pipelines with Handel-C, IEEE Software, 1988, Vol. 150, pp. 39-47.