XII международная конференция

CEE-SECR / РАЗРАБОТКА ПО

28 - 29 октября, Москва

О создании компилятора с высокоуровневого языка на компьютер с программируемой архитектурой

- Борис Штейнберг
- Антон Баглий
- Денис Дубров
- Юрий Михайлуц
- Роман Штейнберг

http://ops.rsu.ru

Цели исследований

1. Сокращение времени разработки программ для систем на кристалле с программируемой архитектурой;
2. Переносимость высокоуровневых программ на программируемые архитектуры с возможным повышением производительности.

Ожидаемый побочный эффект - Стимулирование развития SoC с программируемой архитектурой

Программируемая архитектура вычислительной системы

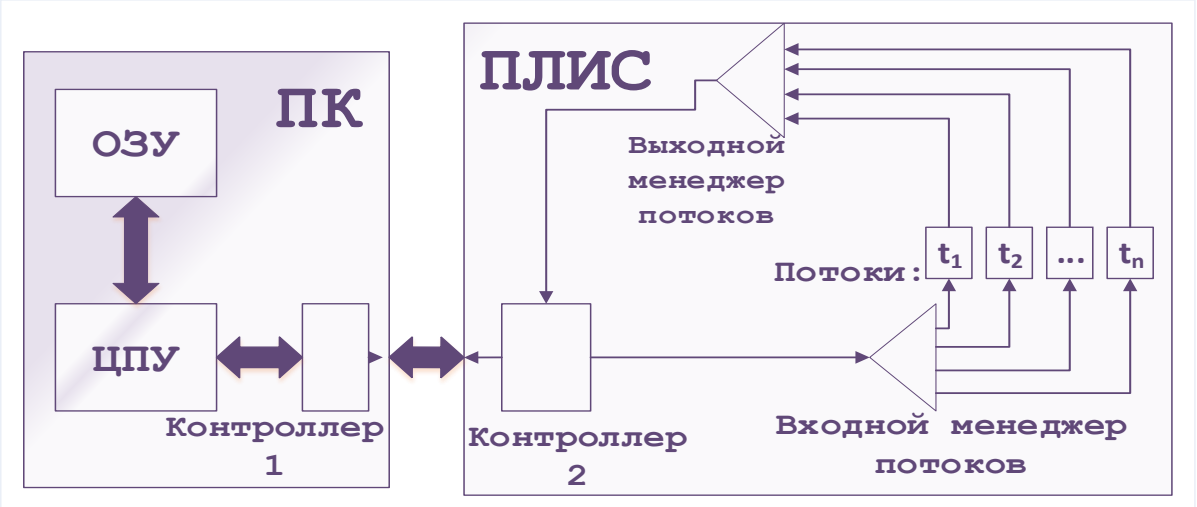
- Настраивается на поддержку каждого конкретного исполняемого алгоритма.
- Более эффективно использует логические ресурсы
- Позволяет гибко распределять логические ресурсы кристалла между вычислительными ядрами (элементарными процессорами) и локальной памятью.
- Позволяет соединять вычислительные ядра (элементарные процессоры) в соответствии со структурой алгоритма.

Компьютер с программируемой архитектурой

ЦПУ и ПЛИС на разных микросхемах

- ➕ Простота реализации.
- ➕ Масштабируемость.
- ➕ Логические ресурсы ПЛИС не тратятся на процессор

- ➖ Низкая скорость обмена ЦПУ и ПЛИС.
- ➖ Потребность в сложном протоколе обмена данными



Реализовано на базе:

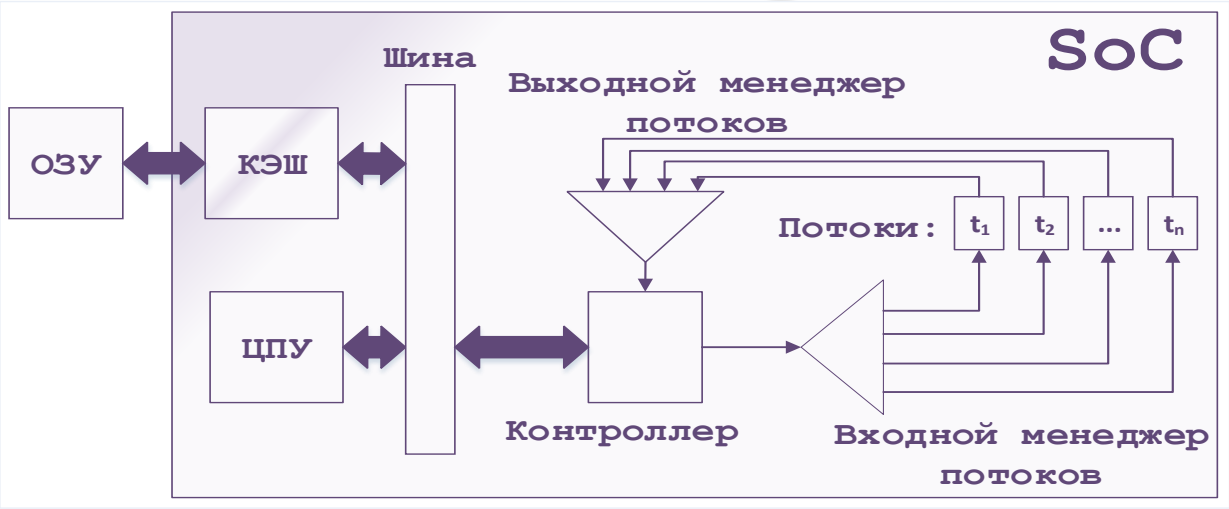
- **ЦПУ:** Core 2 Duo 2,4 MHz
- **ПЛИС:** XC7VX485T в составе комплекта XILINX VC707

Компьютер с программируемой архитектурой

ЦПУ и ПЛИС в одной микросхеме

- ➕ Высокая скорость обмена ЦПУ и ПЛИС.
- ➕ Не требуется сложный протокол обмена данными.

- ➖ Масштабируемость ограничена логическими ресурсами кристалла.



Реализовано на базе:

- **ЦПУ:** soft processor MIPSfpga
- **ПЛИС:** Altera 5CEBA4F23C7N в составе комплекта Terasic DE0-CV

Примеры ускорения алгоритмов при переносе на ПЛИС

```
int main() {
  int a[100], b[100];
  int c[100], d[100];
  int i;
  for (i = 0; i < 100; i ++)
  {
    a[i] = b[i]*c[i] + d[i];
  }
  return 0;
}
```

Параметр	Размер массива		
	10	100	200
Время на ПЛИС, мс	16	84	168
Время на ЦПУ, мс	19	174	339
Отношение	1,1875	2,071	2,017

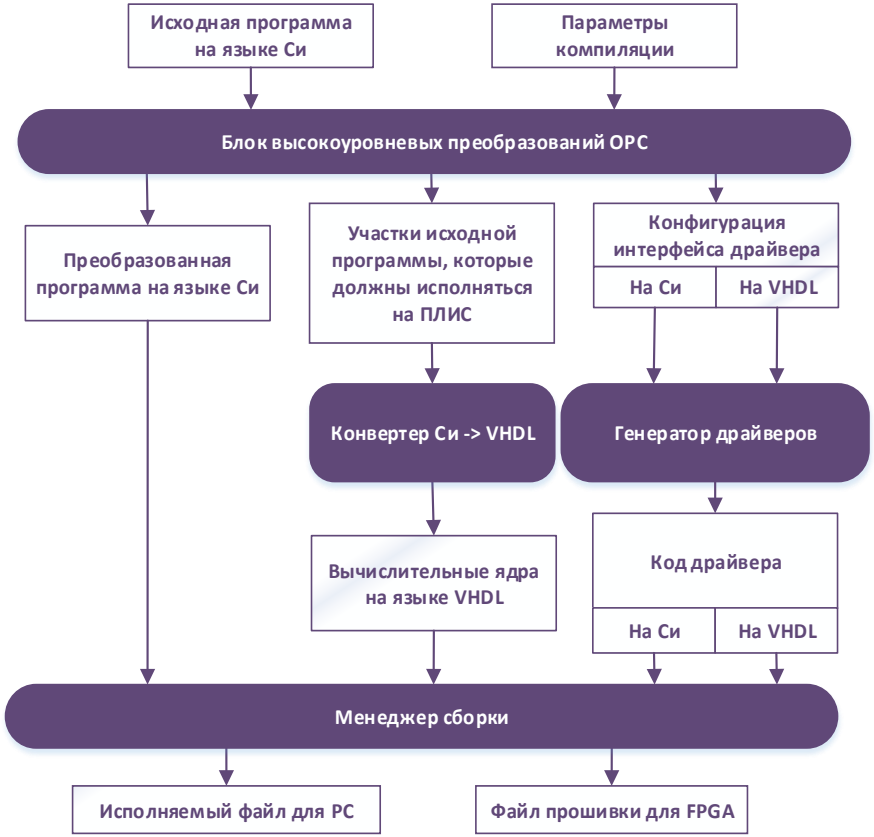
- Эксперимент проводился на отладочной плате Terasic DE0-CV на базе ПЛИС Altera Cyclone V.
- В качестве ЦПУ использовался MIPSfpga, компании Imagination Technologies.

Идея компилятора

1. Находим малую по объему текста, но долго считаемую часть программы.
2. Этой части сопоставляем HDL-код, который компилируем для конкретной ПЛИС
3. В исходной программе заменяем эту часть передачей управления на ПЛИС
4. Возвращаемся к исполнению оставшейся части исходной программы на ЦПУ.

Структура компилятора

Название блока	Выполняемая задача
Блок ОРС	Поиск конвейеризуемых участков кода.
Конвертер Си -> (ВП ОРС)-> VHDL	Преобразование конвейеризуемых участков кода в описание конвейера на VHDL
Генератор драйверов	Генерация вспомогательного кода, обеспечивающего взаимодействие ЦПУ и ПЛИС
Менеджер сборки	Сборка готового проекта из всех преобразованных с сгенерированных файлов.

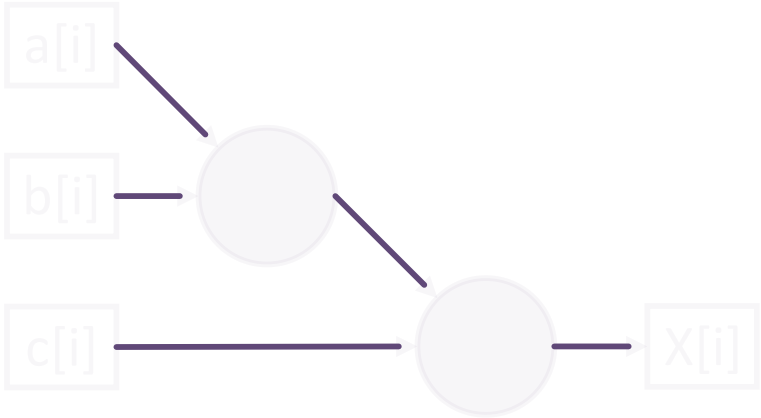


Пример генерация конвейера

Программа на Си

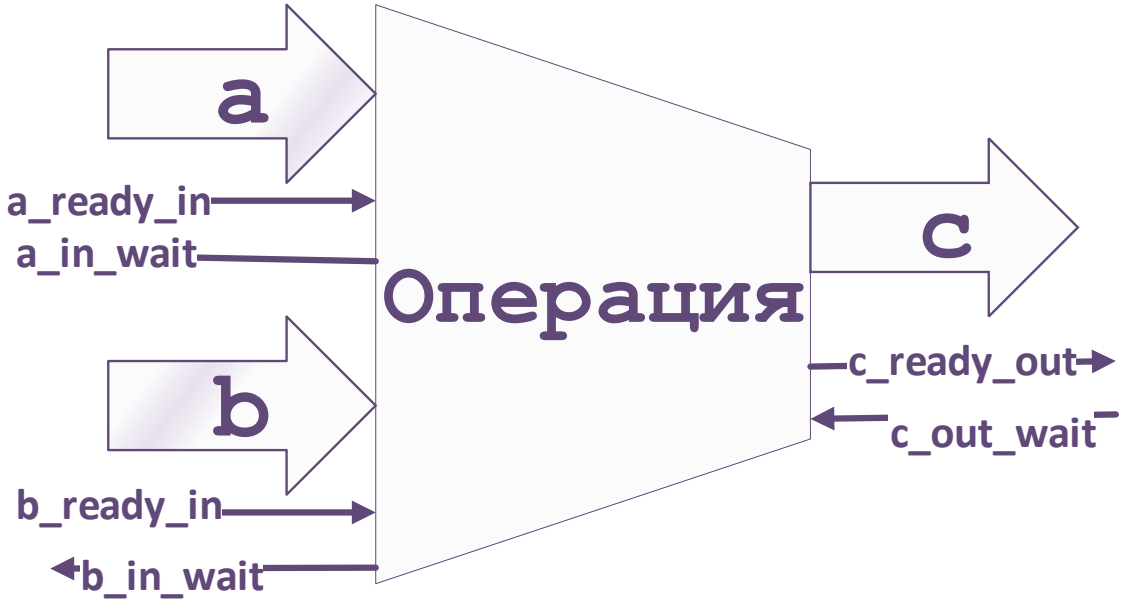
```
for (I = L; I < R; ++ I)
    X[I] = A[I] * B[I] + C[I];
```

Конвейер



Алгоритм синхронизации конвейера

- ✓ Весь VHDL код, сгенерированный компилятором синхронизируется по модели Data flow.



a, b, c	операнды
a_ready_in, b_ready_in, c_ready_out	флаги готовности операндов
a_in_wait, b_in_wait, c_out_wait	флаги ожидания операндов

Аналоги C2HDL

Коммерческие продукты, использующие традиционный язык программирования, такой как C и C++	Catapult C, Vivado Design Suite, Impulse CoDeveloper, Altium Designer и HDL Coder for MATLAB
Академические исследовательские проекты, использующие традиционный язык программирования, такой как C и C++	C-to-Verilog, TCE, Parallel Intellectual Compiler
Продукты, использующие специально разработанные языковые конструкции	Mitrion C, Handel-C и HaSCoL

Благодарим компанию «Imagination Technologies», Юрия Панчула и ООО «Униведа» за предоставление платы «Terasic DE0-CV», а также за предоставление исходных кодов процессора MIPSfpga.

Спасибо за внимание.