

Отзыв на курсовую работу Дмитрия Найданова, посвященную аппаратному ускорению ресурсоемкой задачи из области биоинформатики

Задачей курсовой было ускорение программы, вычисляющей множественное выравнивание строк за счет вынесения самого времяпожирающего куска программы на ПЛИС. В качестве языка аппаратной реализации предлагался HaSCoL. В качестве первого шага было предложено реализовать довольно несложный (порядка 60 строк на C++) алгоритм, ищущий оптимальное относительно целевой функции выравнивание пары строк.

Дмитрий научился писать код на HaSCoL и отлаживать в симуляторе Modelsim сгенерированный VHDL. Само по себе это является довольно сложной технической задачей, требующей понимания как языка HaSCoL, так и языка VHDL. Более того, это требует представления о процессе генерации VHDL по HaSCoL. Усложняет задачу и то, что указанные языки не являются языками программирования в привычном понимании этого слова и требуют понимания того, как устроены и функционируют программируемые логические интегральные схемы (FPGA). Также Дмитрий научился подключать к устройству, описанному на HaSCoL, сторонние блоки памяти на кристалле, описанные на VHDL. Кроме того, он освоил интеграцию симулируемой модели аппаратного устройства с программой на C++ посредством языка SystemC, что позволило ему провести тестирование своего устройства на случайных тестах со сравнением ответов с эталонной функцией, написанной мною на C++.

Минусом работы является ее неполная функциональная корректность: при сравнении аппаратной реализации Дмитрия и программной реализации той же задачи на псевдослучайных тестах, вторая выдает более оптимальный ответ уже на третьем тесте. При этом из исходников аппаратной реализации видно, что алгоритм реализован правильно по модулю немного неправильно заданных граничных условий в методе динамического программирования.

Дополнительным плюсом является то, что Дмитрий изначально написал устройство в таком стиле, что оно сразу отсинтезировалось на частоту в 200МГц. Обычно достижение такой частоты требует нескольких итераций оптимизации.