

ПРОГРАММНОЕ СРЕДСТВО СИНТЕЗА АППАРАТНЫХ УСКОРИТЕЛЕЙ ПО ЗАДАННОМУ ПОЛИНОМУ

Бурко Л. А., Кайки М. Н., Иванюк А. А.

Кафедра информатики, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь

E-mail: {burkoliana, kaikunykhalo}@gmail.com, ivaniuk@bsuir.by

В данной работе рассматривается построение аппаратных ускорителей на основе графов вычислительных процессов. Разработано специализированное программное средство синтеза HDL-описаний аппаратных ускорителей с различными схемотехническими структурами.

ВВЕДЕНИЕ

С каждым годом количество вентилях на кристаллах микросхем непреодолимо растёт, что приводит к увеличению сложности цифровых схем (Систем на Кристалле, Сетей на Кристалле) и увеличению вероятности разработчиками совершать ошибки при проектировании данных структур [1]. Особую роль в современных цифровых устройствах играют аппаратные ускорители, призванные повышать общую производительность системы. Однако с увеличением сложности цифровых устройств возрастает и сложность аппаратных ускорителей, что приводит к большим трудозатратам на их проектирование и верификацию, а также увеличению вероятности появления ошибок в их работе.

I. ОПИСАНИЕ

Разработанное программное средство предназначено для построения аппаратных ускорителей исходя из алгоритма, заданного разработчиком. Помимо алгоритма разработчиком может быть задан набор доступных для использования операций (операционных устройств) со списком их приоритетов друг относительно друга и ограничением на количество применений каждого из них. Также разработчик может выбирать интерфейсы для связи ускорителя с HOST-контроллером или вносить в средство свои способы доступа. После завершения работы средства, разработчик получает набор файлов с HDL-описанием аппаратного ускорителя на выбранном языке описания аппаратуры и набор автоматически сгенерированной документации, содержащей в себе сведения о графе вычислительного процесса, настройках, ограничениях системы, латентности и пропускной способности.

II. ПОЛУЧЕНИЕ ГРАФА ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА

Для построения описания аппаратного ускорителя, предлагаемое средство оперирует графами вычислительного процесса [2], а также их параметрами. Так, для построения графа вычислительного процесса система считывает полином и заданные ограничения (`settings.txt`)

как аргумент при её вызове из командной строки (`agent.py y=a+b+c settings.txt`) и разбивает его на выражения состоящие из одного или двух операндов с использованием обратной польской нотации [3]. Список этих выражений сортируется по количеству вхождений в алгоритм, а приоритет получают наиболее встречаемые. Если имеются ограничения, влияющие на порядок поступления операндов в операционные устройства, уровни в графе, то порядок будет определен по количеству переменных, используемых в выражении. Для выбора оптимального графа исходя из вышеперечисленных ограничений — производится оценка с использованием метрики цикломатической сложности [4], вычисляемой по формуле: $u = e - n + p$, где e — количество ребер, p — количество компонент связности, n — количество узлов. Исходя из полученных значений для каждого из вариантов графа решается задача нахождения минимума значения цикломатической сложности. Стоит отметить, что на данном этапе граф вычислительного процесса может быть изменён разработчиком с целью внесения правок в алгоритм или оптимизации его работы вручную.

После получения графа, производится процесс генерации HDL-описания аппаратного ускорителя. Метод, которым описывается структура ускорителя на HDL-языке может быть задан разработчиком или выбран из присутствующих в разработанном средстве. Например, аппаратный ускоритель может быть построен в виде таких структур:

1. Однотактный ускоритель.
2. Конвейерный ускоритель.
3. Конечный автомат.
4. Микропрограммный автомат.
5. Микропроцессорная система со специфической системой команд.

III. ПРИМЕР

В качестве примера, рассмотрим построение и синтез аппаратного ускорителя для заданного полинома: $y = x \cdot a \cdot b + 3 \cdot a \cdot b + c$ (`agent.py y=abx+3ab+c settings.txt`). Произведём синтез двух графов вычислительного процесса (см. рис. 1). Первый граф получим при

неограниченном количестве аппаратных ресурсов (умножителей, сумматоров). При таком подходе расчёт выражений $x \cdot a \cdot b$ и $3 \cdot a \cdot b$ производится параллельно с использованием четырех умножителей и двух сумматоров. Для получения второго графа введём такие ограничения — граф должен обладать наименьшим возможным количеством стадий и операций на них. Выражение $y = x \cdot a \cdot b + 3 \cdot a \cdot b + c$ подлежит упрощению — средство способно вынести множитель ab за скобки, тогда выражение принимает такой вид: $y = ab \cdot (x+3) + c$. В таком случае снижается количество необходимых умножителей, сумматоров.

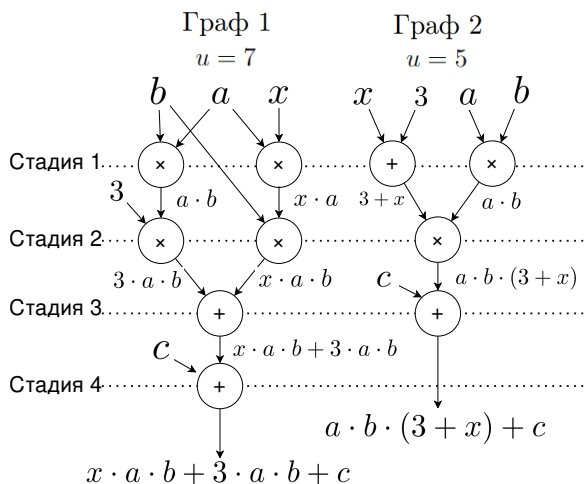


Рис. 1 – Графы вычислительного процесса для выбранного полинома

После получения графов вычислительного процесса сгенерируем четыре HDL-описания аппаратных ускорителей с последующим процессом размещения и трассировки на кристалле ПЛИС:

1. Однотактовая структура графа 1.
2. Однотактовая структура графа 2.
3. Конвейерная структура графа 1.
4. Конвейерная структура графа 2.

Однотактовая структура подразумевает установку регистров только на входах и выходах схемы, конвейерная — на входах, выходах и между стадиями графа (см. рис. 1).

В качестве интерфейса для связи с HOST-контроллером выберем AXI4-Lite и укажем генератору на необходимость внедрить контроллер данного интерфейса в структуру ускорителя. В результате работы разработанного средства получим наборы файлов с Verilog-кодом для каждой из структур. С целью оценки аппаратных затрат на реализацию каждого из ускорителей, был проведён процесс размещения и трассировки этих структур на кристалле ПЛИС компании Xilinx. Для проведения процесса синтеза был спроектирован тестовый стенд, содержащий в себе софт-процессор Microblaze с AXI-интерконнектом для доступа к регистрам аппа-

ратного ускорителя. Тип аппаратного ускорителя выбирался при помощи определения констант проектирования и конструкции generate. Процесс размещения и трассировки проводился для кристалла ПЛИС семейства Artix-7 — xc7a100tscg384-1. В качестве ограничений по частоте было выбрано значение 100 МГц. В результате работы САПР Vivado были получены отчёты по затрачиваемым ресурсам и задержкам схемы на кристалле. Данные из отчётов системы проектирования а также расчётная производительность (см. формулу 2) — приведены в таблице 1.

$$P = \frac{f_{CLK} \cdot L}{S}, [10^6 \cdot \text{Операций/с}] \quad (1)$$

где f_{CLK} — максимальная частота синхронизации, L — число тактов синхронизации для выполнения одной операции при полной загрузке конвейера, S — число тактов синхронизации для выполнения одной операции при нулевой загрузке конвейера.

Таблица 1 – Аппаратные затраты и производительность схем ДКП

Ресурсы	Структура			
	1	2	3	4
LUTs	209	127	193	111
FF	64	64	224	168
f_{CLK} [МГц]	51,414	55,839	112,878	112,713
L	1	1	1	1
S	1	1	4	3
P	51,41	55,83	28,21	37,57

Анализируя полученные данные (табл. 1), можем сделать такие выводы: Граф №2 в обоих случаях показал большую производительность и потребовал меньше ресурсов для размещения в кристалле ПЛИС.

IV. Вывод

Разработанное средство для синтеза аппаратных ускорителей по заданному полиному позволяет не только быстро получать HDL-описания аппаратных ускорителей для вычисления полиномов, но и проводить их оптимизацию с целью повышения характеристик быстродействия и уменьшения аппаратных затрат для их размещения на кристалле цифровых микросхем.

1. Denning Peter J. Exponential Laws of Computing Growth Communications of the ACM / Peter J. Denning, Ted G. Lewis, — January 2017, Vol. 60 No. 1, — P. 54-65
2. Мак-Кракен Д. Численные методы и программирование на Фортране / Д. Мак-Кракен, У.Дорн. Москва «Мир», 1977, — 79-83с.
3. Пратт, Т. Языки программирования. Разработка и реализация / Т. Пратт, М. Зелковиц. — Изд. 4-е, 2002, — 336-343с.
4. McCabe. A Complexity Measure / IEEE Transactions on Software Engineering: journal. — 1976. — December. — P. 308–320.