

АППАРАТНЫЕ АЛГОРИТМЫ ВЫЧИСЛЕНИЯ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ

Д.А. Городецкий

Национальная академия наук Беларуси, Минск, Беларусь
danila.gorodecky@gmail.com

В докладе предлагается подход к проектированию элементов, выполняющих арифметические операции для целых чисел и чисел с плавающей запятой. Подход основан на двух этапах.

На одном этапе осуществляется разбиение входных данных (т.е. чисел) на наборы меньших разрядностей. Разрядность наборов, на которые разбиваются числа, зависит от технических особенностей элементной базы, которую планируется использовать для аппаратной реализации алгоритма вычисления. Например, если проектируется устройство для реализации на FPGA, и базовые вычислительные ячейки – таблицы состояний (LUTs) – имеют четыре входа, то числа целесообразно разбивать на 4-битные наборы. Если проектируется устройство для реализации на заказной схеме (ASIC), состоящей из библиотечных элементов, то разрядность наборов следует подбирать под технические особенности библиотеки элементов ASIC.

Экспериментальные исследования показали, что при реализации умножителей двух натуральных чисел, разрядность которых не превышает 32 бита, наиболее эффективным (с точки зрения быстродействия) образом является разбиение чисел на 4-битные наборы [1]. При реализации аппаратного алгоритма нахождения остатка от деления натуральных чисел эффективным (с точки зрения быстродействия и аппаратных затрат) является разделение делимого (разрядность которого может достигать более 4 тысяч бит) на наборы, разрядность которых не превышает 12 бит [2].

В зависимости от арифметической операции, наборы, на которые разбиваются входные числа, рассматриваются как друг от друга независимые операнды, над которыми отдельно или попарно производится соответствующие арифметические операции. Например, при рассмотрении двухоперандной операции умножения 8-битных чисел $A \cdot B = R$, каждое из чисел разбивается на два 4-битных операнда, т.е. $A = (A_2, A_1)$ и $B = (B_2, B_1)$, где $A_1 = (a_4, a_3, a_2, a_1)$, $A_2 = (a_8, a_7, a_6, a_5)$, $B_1 = (b_4, b_3, b_2, b_1)$ и $B_2 = (b_8, b_7, b_6, b_5)$. В этом случае $A \cdot B = A_1 \cdot B_1 + A_1 \cdot B_2 \cdot 2^4 + A_2 \cdot B_1 \cdot 2^4 + A_2 \cdot B_2 \cdot 2^8 = R_1 + 2^4 \cdot R_2 + 2^4 \cdot R_3 + 2^8 \cdot R_4$. Рассматривая операцию нахождения остатка от деления 8-битного числа X на число 13, где $X = (X_2, X_1)$, $X_1 = (x_4, x_3, x_2, x_1)$ и $X_2 = (x_8, x_7, x_6, x_5)$, результат вычислений может быть представлен следующим образом:
$$X(mod\ 13) = X_1(mod\ 13) + (2^4 \cdot X_2)(mod\ 13) = X_1(mod\ 13) + (3 \cdot X_2)(mod\ 13) = (S_1 + S_2)(mod\ 13).$$

Суть другого этапа предлагаемого подхода состоит в рассмотрении промежуточных результатов вычислений в качестве систем булевых функций с целью минимизации числа слагаемых (например, конъюнкций) или построения оптимальной диаграммы двоичного выбора. Для приведенных примеров умножения и нахождения остатка от деления составляются системы булевых функций для промежуточных результатов R_1, R_2, R_3, R_4, S_1 и S_2 .

На этом этапе является целесообразным учитывать свойства булевых функций, которые описывают рассматриваемую арифметическую операцию. Например, применение свойств симметрии булевых функций при рассмотрении коммутативных арифметических операций позволяет до трех раз сократить число библиотечных элементов, основанных на использовании нормальных форм (дизъюнктивных, полиномиальных и т.д.) [3].

Предлагаемый подход позволяет до 300% сократить аппаратные затраты и увеличить скорость обработки информации по сравнению с подходами, используемыми современными промышленными САПР.

Литература

1. Gorodecky D. *Design of Multipliers using Fourier Transformation // Further Improvements in the Boolean Domain / Cambridge Scholar Publishing, UK. – Section 3.4, pp. 240-252.*
2. Gorodecky D. and Villa T. *Efficient Hardware Operations for the Residue Number System by Boolean Minimization // Advanced Boolean Techniques / Springer. – Section 11, pp.237-258.*
3. Gorodecky D. *Reed-Muller Representation n Arithmetic Operations // Reed-Muller Workshop, May 24, 2019, Fredericton, New Brunswick, Canada. – pp. 53-58.*

ARITHMETIC OPERATIONS IN HARDWARE

D.A. Gorodecky

National academy of sciences of Belarus, Minsk, Belarus