

FPGA-Systems 2020

Многоразрядные умножители в ПЛИС

Несмотря на широкое распространение ПЛИС со встроенными DSP-блоками, в том числе повышенной разрядности (до 48 бит), ряд задач требуют выполнения вычислений с еще большей разрядностью. Кроме того, в приложениях, критичных к потребляемой мощности, или обновлении старых платформ часто приходится использовать ПЛИС без встроенных умножителей. В этом случае умножители приходится делать самим.

Краткое введение.

- Многоразрядным (для определенности) будем считать умножитель с разрядностью выше типовых 18×18 ... 36×36 бит.
- По практическим соображениям, многоразрядные умножители, как правило, реализуются последовательностными, то есть с многократным суммированием частичных произведений (в пределе — бит-последовательными). То есть, «длинные» числа обрабатываются «по частям» («Divide and conquer»).
- Образование частичных произведений — не очень вариативная вещь, т.к. основных вариантов по сути всего три: использование встроенных умножителей, прямое (побитное) маскирование и использование канонической знакоразрядной формы (алгоритм Бута-Лемана).
- Гораздо большее поле для компромиссов затраты/эффективность появляется при построении сумматоров частичных произведений. Поэтому именно вопросам их реализации до сих пор и посвящено основное число работ по проектированию умножителей.

Краткое введение – продолжение.

- В случае, когда частичные произведения имеют разрядность больше нескольких бит, имеет смысл использовать следующий способ сокращения их числа (N – размерность подмножителей):
- Пусть $Z = (A \cdot 2^N + B) \cdot (C \cdot 2^N + D)$. Формально $Z = A \cdot C \cdot 2^{2N} + (A \cdot D + B \cdot C) \cdot 2^N + B \cdot D$.
- Однако $A \cdot D + B \cdot C = (A+B) \cdot (C+D) - (A \cdot C + B \cdot D)$. То есть, если мы располагаем умножителем с разрядностью $N+1 \times N+1$, то мы можем обойтись тремя умножениями вместо четырех для формирования произведения разрядности $4 \cdot N$ (похожий прием может использоваться и при умножении комплексных чисел). Этот **метод Карацубы** при рекурсивном применении позволяет уменьшить количество частичных умножений с $O(n^2)$ до $O(n^{1.59})$.
- Суммирование частичных произведений может выполняться как «в лоб» (см. «умножение столбиком», именно так его построит синтезатор при отсутствии дополнительных констрейнов), так и более интеллектуально.

Пример карты частичных произведений

– A B
– C D

– _____
 • B*D
– B*C
– A*D
– A*C

Краткое введение – продолжение.

- Суммирование частичных произведений в последовательностном умножителе, как правило, имеет смысл делать с запоминанием переносов, подобно тому, как выполняются многобайтные суммирования/вычитания в восьмибитных процессорах. Это позволяет избежать применения медленных «длинных» сумматоров.
- Порядок выполнения суммирования частичных произведений также влияет на сложность автомата управления и количество распределенной памяти. Наиболее компактен «итальянский» метод (или метод палочек Непера) – суммирование «по косым полосам справа», в этом случае необходимая разрядность сумматора-аккумулятора лишь на $\log_2(M)$ больше разрядности частичных произведений.
- В ряде случаев большая разрядность нужна только для одного из сомножителей, например, при реализации IIR интеграторов/корректоров в цепях управления с очень низкими (относительно частоты квантования) резонансными частотами.

Основные отличия ПЛИС от ASIC

- Особенности ПЛИС диктуют иные подходы к оптимизации, чем при проектировании ASIC, особенно велик разрыв при «ручной» топологии ASIC, например, спроектированный «вручную» 64-битный сумматор на древнем 0.5 мкм процессе имеет задержку менее 1 нс, чего до сих пор толком не могут обеспечить ПЛИС;
- Существенно разная «цена» логических функций и каналов трассировки: в ПЛИС задержки на коммутируемых связях существенно больше, чем в ASIC, соответственно, локальность связей имеет приоритет над «сложностью» логики даже в ПЛИС с простыми 4-входовыми LUT;
- Неэффективная реализация мультиплексоров с числом входов больше 2, как в плане расхода ресурсов, так и быстродействия;
- Нужно различать ситуации, когда требуется максимальная общая производительность на данном количестве элементов и максимальная производительность отдельного узла, без учета занимаемых им ресурсов.

Многотактные умножители для ПЛИС

- «Классический» многотактный умножитель, со сдвигом вправо на один бит за такт.
- Модификация с использованием алгоритма Бута-Лемана (2 бита за такт).
- Умножитель на сумматоре с запоминанием переноса и бит-последовательным выходом (см. Лайон). Предельно лаконичен, нет длинных цепей — как правило, тактовая частота лимитируется только возможностями глобальной тактовой сети.
- Умножитель на основе многовходового последовательного сумматора. По общим свойствам он близок к предыдущему, но из-за другой структуры распространения переносов более гибкий.

Рекомендуемая литература по Bit-serial

- US Pat. № 6584481B1. Xilinx.
- Paolo Ienne and Marc A. Vireliz, Bit-serial Multipliers and Squarers, IEEE Transactions on Computers, 43(12) December 1994, p. 1445-1450 - Хороший краткий обзор.
- R.F. Lyon. Two's complement pipeline multipliers. IEEE Trans. On Communications, COM-24(4),418-425. April 1976. Классическая работа.
- Paraskevas Kalivas, Kiamal Pekmestzi, Paul Bougas, Andreas Tsiricos and Kostas Gotsis, «Low-Latency and High-Efficiency Bit-serial Multipliers». Очень наглядно и доходчиво описана вполне пригодная структура.
- Kelly Johansson's dissertation, «Low complexity and low power Bit-serial Multipliers», Linköping, 2003. Эффективные умножения на фиксированные коэффициенты и применение правила Горнера для повышения точности.