



УДК 004.312.44

© 2006 г. **Н.Ю. Сорокин**

(Тихоокеанский государственный университет, Хабаровск)

МОДУЛИ ДЕЛЕНИЯ НА ПЛИС

В работе приводятся результаты создания высокоскоростных модулей целочисленного деления на ПЛИС. Проведен сравнительный анализ имплементации различных алгоритмов деления со стандартным модулем фирмы Xilinx. Показано, что реализованные модули по скорости и разрядности значительно превосходят модуль целочисленного деления от Xilinx.

Введение

В настоящее время особой популярностью пользуются ПЛИС (FPGA) для проектирования сложных систем. При этом решается большое количество задач, направленных на получение повышенной точности конечных либо промежуточных результатов, а также таких на решение которых накладываются определенные ограничения.

Для упрощения проектируемых устройств чаще всего применяется целочисленная арифметика, так как использование модулей с плавающей точкой не всегда оправдано и приводит к значительному усложнению устройств. Однако существующие в настоящее время реализации алгоритмов целочисленного деления накладывают дополнительные ограничения в процессе разработки вычислительных модулей на базе программируемой логики. Такими ограничениями являются представления операндов и результата деления. Например, в стандартном программном обеспечении для Xilinx FPGA предлагается использовать блок целочисленного деления со следующими параметрами: 32-битные операнды, 32-битный частичный остаток от деления. Для решения задач с повышенной точностью результата этого модуля недостаточно.

Еще одна особенность заключается в том, что данный модуль не является быстродействующим и занимает достаточно большую площадь на кристалле.

Целью данной работы являются разработка быстродействующих модулей целочисленного деления, получение неограниченной разрядности остатка от деления и оптимизация этих модулей для Xilinx FPGA.

Методы целочисленного деления

Как известно, основу целочисленного деления составляет общепринятый способ деления с помощью операций вычитания или сложения и сдвига. Задача сводится к вычислению частного $Q \in Z_{\geq 0}$ и остатка $S \in Z_{\geq 0}$ от деления $Y \in Z_{\geq 0}$ на $D \in Z_{>0}$ – таких, что $Q = \text{int}(Y/D)$ и $S = Y - Q \cdot D$ при $S < D$. Деление выражается как последовательность вычитаний делителя сначала из делимого, а затем из образующихся в процессе деления частичных остатков.

При реализации на уровне логических схем делимое

$$Y(y_{2n-1} y_{2n-2} \dots y_1 y_0)$$

обычно представляется двойным словом ($2n$ разрядов, $n \in Z_{>0}$), делитель

$$D(d_{n-1} d_{n-2} \dots d_1 d_0),$$

частное

$$Q(q_{n-1} q_{n-2} \dots q_1 q_0)$$

и остаток

$$S(s_{n-1} s_{n-2} \dots s_1 s_0)$$

имеют разрядность n . В данной работе рассматриваются только положительные (беззнаковые) числа, что не является значительным ограничением применения результатов работы.

Операция деления выполняется за n итераций и описывается следующим образом:

$$S^{(i)} = 2S^{(i-1)} - q_{n-i}(2^n D), \quad (1)$$

при

$$S^{(0)} = Y, \quad S^{(n)} = 2^n S \quad \text{и} \quad q_{n-i} = \begin{cases} 1, & \text{если } (2S^{(i-1)} - 2^n D) \geq 0, \\ 0, & \text{если } (2S^{(i-1)} - 2^n D) < 0. \end{cases}$$

После n итераций, на основании приведенных формул, получаем

$$S^{(n)} = 2^n S^{(0)} - Q(2^n D) = 2^n [Y - (Q \times D)] = 2^n S. \quad (2)$$

Частное от деления $2n$ -разрядного числа на n -разрядное может содержать более чем n разрядов. В этом случае возникает переполнение, из-за чего перед выполнением деления необходима проверка условия $Y < 2^n D$.

Наиболее очевидный алгоритм целочисленного деления носит название алгоритма деления с неподвижным делителем и восстановлением остатка (алгоритм 1).

Недостаток алгоритма деления с восстановлением остатка заключается в необходимости выполнения на отдельных шагах дополнительных операций сложения для восстановления частичного остатка. Это увеличивает время выполнения деления, которое в таком случае может меняться в зависимости от конкретного сочетания кодов операндов.

В силу указанных причин реальные делители строятся на основе алгоритма деления с неподвижным делителем без восстановления остатка (алгоритм 2) [1, 2].

Алгоритм 1. Деление с восстановлением остатка

1. Исходное значение частичного остатка (ЧО) полагается равным старшим разрядам делимого.
2. ЧО удваивается путем сдвига на один разряд влево. При этом в освобождающийся при сдвиге младший разряд ЧО заносится очередная цифра частного.
3. Из сдвинутого ЧО вычитается делитель и анализируется знак результата вычитания.
4. Очередная цифра модуля частного равна единице, когда результат вычитания положителен, и нулю, если отрицателен. В последнем случае значение остатка восстанавливается до того значения, которое было до вычитания.
5. Пункты 2-4 выполняются последовательно для получения всех цифр модуля частного.

Алгоритм 2. Деление без восстановления остатка

1. Исходное значение ЧО полагается равным старшим разрядам делимого.
2. Частичный остаток удваивается путем сдвига на один разряд влево. При этом в освобождающийся при сдвиге младший разряд ЧО заносится очередная цифра частного.
3. Из сдвинутого ЧО вычитается делитель, если остаток положителен, и к сдвинутому частичному остатку прибавляется делитель, если остаток отрицателен.
4. Очередная цифра модуля частного равна единице, когда результат вычитания положителен, и нулю, если он отрицателен.
5. Пункты 2-4 выполняются последовательно для получения всех цифр модуля частного.

Методы ускорения операции деления

Оптимизация операции целочисленного деления с целью повышения скорости деления – достаточно трудоемкий процесс. Тем не менее определенные возможности для ускорения деления существуют, и их можно свести к следующим:

замена делителя обратной величиной, с последующим ее умножением на делимое;

сокращение времени вычисления частичных остатков в традицион-

ных методах деления (с восстановлением или без восстановления остатка) за счет ускорения операций суммирования (вычитания);

сокращение времени вычисления за счет уменьшения количества операций суммирования (вычитания) при расчете значения частного остатка;

вычисление частного в избыточной системе счисления.

За исключением первого из перечисленных подходов, все прочие фактически являются модификациями традиционного способа деления.

Операцию умножения можно производить сравнительно быстро, если применить комбинационные схемы параллельного умножения. Данное обстоятельство можно использовать, заменив операцию деления умножением на обратную величину $1/D$, и, соответственно, свести задачу к эффективному вычислению значения $1/D$. Обычно задача решается одним из трех методов: с помощью ряда Тейлора, метода Ньютона-Рафсона и при использовании свойства нечетных чисел [3, 4]. Замена операции деления на умножение более характерна для чисел с плавающей запятой, что в основном и используется для реализации операции деления в современной микропроцессорной технике.

Ускорение вычисления частичных остатков достигается путем применения быстрых схем сложения, при использовании различных приемов ускорения распространения переноса. Также часто применяются матричные схемы сложения, являющиеся регулярными и удобными для реализации в виде интегральной микросхемы. Такие же схемы используются и при реализации на кристалле быстрых схем умножения [3, 5].

В основе третьей группы методов ускорения операции деления лежит так называемый алгоритм SRT, различные реализации которого описаны в работах [6 – 8]. Этот алгоритм представляет собой модификацию деления без восстановления остатка. В стандартной процедуре на каждом шаге, помимо сдвига частичного остатка, производится прибавление либо вычитание делителя. В SRT-алгоритме сдвиг частичного остатка также имеется в каждой итерации, однако сложение или вычитание в зависимости от получающегося частного остатка на отдельных шагах может не выполняться, что безусловно позитивно влияет на быстродействие деления.

Рассмотрим данный алгоритм применительно к положительным целым числам. Делимое представляется n -разрядным числом, а делитель – m -разрядным. Процедура деления начинается с удаления в делителе всех нулей, предшествующих старшей единице, реализуемой при помощи сдвига делителя влево на требуемое число разрядов. На аналогичное число разрядов влево сдвигается и делимое.

Далее выполняются итерации, в которых вычисляются цифры частного и частичные остатки. Действия, выполняемые на i -й итерации, можно описать следующим образом:

$$q_i = \begin{cases} 1, & \text{если } 2S^{(i-1)} \geq D, \\ 0, & \text{если } -D \leq 2S^{(i-1)} < D, \\ -1, & \text{если } 2S^{(i-1)} < -D, \end{cases} \quad (3)$$

$$S^{(i)} = 2S^{(i-1)} - q_i D. \quad (4)$$

В формуле (4) частное представляется в системе счисления, отличной от двоичной. Это означает, что цифры частного могут иметь больше чем два значения $\{0,1\}$. В рассматриваемом случае – $\{-1,0,1\}$. По завершении всех итераций, если последний остаток отрицателен, выполняется коррекция этого остатка и полученного частного, для чего к остатку прибавляется делитель, а из частного вычитается единица с весом младшего разряда. Последний шаг в алгоритме — преобразование частного из системы $\{-1,0,1\}$ в систему $\{0,1\}$.

Необходимо отметить, что представление чисел в избыточных системах исчисления, – например, в алгоритме умножения Бута [3, 5] – является одним из самых эффективных методов повышения производительности. В то же время реализация такого подхода ведет к усложнению аппаратуры делителя, в частности надстраивается логика определения операции, выполняемой в очередной итерации. Для этой цели в состав устройства деления включается специальная память, хранящая таблицу, определяющую необходимые действия, в зависимости от текущей комбинации цифр в частичном остатке и делителе. Тем не менее выигрыш в быстродействии оказывается решающим моментом, и данный метод широко используется в микропроцессорах.

Устройства целочисленного деления

Вопросами создания аппаратных целочисленных устройств деления ученые занимаются достаточно давно. Одними из основополагающих работ были труды, описанные в источниках [2, 7]. Полные доказательства корректности работы алгоритмов RNS были даны в [2], однако первая предложенная схема не могла быть использована для получения большого числа бит остатка от деления, а вторая схема использовала взвешенные дробные числа, что проводило к значительному усложнению устройства и снижало производительность. Другое решение целочисленного модуля деления было представлено, например, в работе [9], а имплементация на базе алгоритма SRT – в [6]. Несмотря на достаточное количество реализаций модулей целочисленного деления, оптимизация имплементаций является одним из основных моментов исследований, – например, выполнение операции деления без дополнительного масштабирования входных операндов, как в [9]. В противоположность предыдущим работам данное исследование ориентированно на создание модулей целочисленного деления, выпол-

няющих операцию деления с целыми числами, разрядность которых превышает 64 и 128 бит. Частично результаты представлены в статье [10].

К современным устройствам деления можно отнести стандартный модуль целочисленного деления IP Core Xilinx Divider v3.0. Данный модуль представляет собой параметризованное синхронное устройство, позволяющее делить n -разрядное делимое на m -разрядный делитель. Результат деления представляется в виде пары чисел – частное и остаток. Основные свойства модуля деления фирмы Xilinx:

использование в качестве модуля (IP core) для всех семейств Virtex™, Virtex-E, Virtex-II™, Virtex-II Pro™, Spartan™-II, Spartan-III, Spartan-3 и Virtex-4 FPGA;

разрядность делимого – от 1 до 32 бит;

разрядность делителя – от 3 до 32 бит;

разрядность дробного остатка (fractional remainder) – от 3 до 32 бит.

В табл. 1 представлены характеристики модуля целочисленного деления фирмы Xilinx. Этот модуль имеет два значительных недостатка. Во-первых, занимает большое место на кристалле, а во-вторых, у него достаточно ограниченные возможности использования – значения остатка от деления не более 32 бит, что не позволяет использовать этот модуль как универсальный во всех вычислительных задачах, решаемых на базе ПЛИС.

Таблица 1

Характеристика	Разрядность остатка, бит		
	8	16	32
Количество секций (slices)	2247	2742	3843
Количество триггеров (Flip Flops)	4020	4904	6864
Количество таблиц (LUTs)	1400	1680	2240
F_{\max} , МГц	204,3	201,6	193,1

Реализация модулей целочисленного деления

Для реализации на базе Xilinx FPGA были выбраны алгоритмы целочисленного деления с восстановлением и без восстановления остатка, а также модификация алгоритма SRT.

Все алгоритмы были закодированы с помощью языка VHDL без использования каких-либо специфических конструкций и блоков, – например, сумматоров, умножителей Xilinx FPGA. Это позволяет использовать данные разработки (VHDL код) в разработках, ориентированных не только на применение микросхем программируемой логики фирмы Xilinx, но и, например, микросхем фирмы Altera. При этом все характеристики модулей деления для сравнения были взяты только с проектов (дизайнов), выполненных на микросхемах Xilinx FPGA Virtex-II Pro™. Реализация дизайнов выполнена с использованием САПР Xilinx ISE 6.3i. Верификация реализованных модулей целочисленного деления произведена с помощью симуля-

тора ModelSIM XE 5.8.

Все разработанные модули целочисленного деления имеют одинаковый интерфейс, состоящий из входных шин операндов (n -разрядное делимое и m -разрядный делитель), выходной шины q -разрядного результата и шины управления, включающей синхросигнал, сигналы сброса, запуска, готовности и ошибки. Значения разрядности каждой шины объявлены как параметры модуля (с помощью конструкции *generic*), что позволяет полностью параметризовать код модуля. При этом количество циклов, необходимых для вычисления q -разрядного частного, равно q плюс циклы инициализации для каждого разработанного модуля деления.

При реализации модулей значительные усилия были затрачены на оптимизацию. Помимо конвейеризации, значительную роль в повышении быстродействия сыграла оптимизация с помощью временных ограничений (timing constraints) и ограничений по площади (area constraints) при расположении дизайнов на кристалле. Наложение условий по времени на синхросигнал, а также на некоторые из внутренних сигналов позволило оптимизировать дизайн по быстродействию. Ограничения по площади были использованы также для увеличения быстродействия дизайнов за счет их расположения вокруг некоторого набора аппаратных ресурсов – сумматоров и умножителей.

Результаты проектирования модулей

Целью работы являлось получение модулей целочисленного деления для ПЛИС, характеристики которых, – например, разрядность входных и выходных операндов – превосходили бы стандартный модуль деления фирмы Xilinx. Параметризованные модули были спроектированы для разрядностей входных и выходных операндов от 8 до 128 бит, каждый из них был синтезирован и размещен на ПЛИС фирмы Xilinx. В результате реализации получены данные о скорости и занимаемой площади на кристалле. Среди фиксируемых характеристик были следующие: максимальная частота работы дизайна, количество занимаемых секций (slices) на микросхеме, количество используемых триггеров, таблиц (LUTs). Вся эта информация собиралась для различных комбинаций разрядностей входных и выходных операндов. Результаты анализа представлены в работе [10].

Сравнительные характеристики для модулей с различным количеством разрядов входных данных представлены в табл. 2 – 5. Как видно из представленных результатов, модуль деления без восстановления остатка значительно превосходит по скорости и значительно ниже по занимаемой площади, чем стандартный делитель фирмы Xilinx. Ввиду особенностей алгоритма при реализации была проведена значительная оптимизация вычислительной структуры, за счет чего получена высокая скорость вычислений. Делитель с восстановлением остатка отстает по временным пара-

метрам, но по аппаратным затратам практически не уступает делителю без восстановления остатка [10].

Таблица 2

Характеристики модулей деления с 32-разрядными входными операндами*

Модуль	Разрядность остатка, бит				
	8	16	32	64	128
IP Core Xilinx Divider v3.0	204,3 / 211	201,6 / 253	193,1 / 350	–	–
Алгоритм SRT	78,1 / 525	77,2 / 635	76,1 / 854	72,4 / 1340	64,3 / 2504
Алгоритм с восстановлением остатка	126,6 / 324	120,9 / 405	108,8 / 597	89,1 / 1089	81,2 / 1983
Алгоритм без восстановления остатка	248,6 / 165	247,4 / 198	244,9 / 265	193,1 / 502	180,3 / 893

Таблица 3

Характеристики модулей деления с 64 разрядными входными операндами

Модуль	Разрядность остатка, бит		
	32	64	128
IP Core Xilinx Divider v3.0	<i>не поддерживается</i>		
Алгоритм SRT	62,5 / 1552	61,8 / 2087	60,5 / 3190
Алгоритм с восстановлением остатка	75,3 / 1288	57,8 / 2232	48,4 / 3988
Алгоритм без восстановления остатка	173,6 / 559	160,3 / 805	157,6 / 1225

Таблица 4

Характеристики модулей деления с 128-разрядными входными операндами

Модуль	Разрядность остатка, бит		
	32	64	128
IP Core Xilinx Divider v3.0	<i>не поддерживается</i>		
Алгоритм SRT	47,9 / 3361	47,1 / 4098	46,2 / 5563
Алгоритм с восстановлением остатка	42,8 / 3762	38,9 / 4961	36,6 / 7022
Алгоритм без восстановления остатка	92,2 / 1746	89,9 / 2147	86,5 / 2971

* Здесь и далее размерность дроби – (МГц/нс).

Заключение

Стандартный модуль деления для ПЛИС фирмы Xilinx не предоставляет возможность получения остатка от деления более чем 32 разряда, что значительно ограничивает точность вычислений. Так как итогом подобной операции является в общем случае приближенное значение, то это может привести к неудовлетворительным результатам при использовании такого модуля. В данной работе разработка и последующая оптимизация модулей целочисленного деления показала значительное превосходство по скорости модуля деления без восстановления остатка, причем данный модуль является более универсальным, чем стандартный модуль фирмы Xilinx.

ЛИТЕРАТУРА

1. *Цилькер Б.Я., Орлов С.А.* Организация ЭВМ и систем. – СПб.: Питер, 2004.
2. *Hiasat A.A., Abdel-Aty-Zohdy H.S.* Semi-Custom VLSI Design and Implementation of a New Efficient RNS Division Algorithm // *The Computer Journal.* – 1999. – № 3(42). – P.232-240.
3. *Mueller S.M., Paul W.J.* Computer Architecture Complexity and Correctness. – Springer-Verlag, 2000.
4. Parhami Behrooz *Computer Arithmetic: Algorithms and Hardware Designs.* – N.Y.: Oxford University Press, 2000.
5. *Paul W. J., Seidel P.-M.* To Booth or not to Booth // *Integration the VLSI journal.* – 2002. – № 1-3(32). – P.5-40.
6. *Harris D.L., Oberman S.F., Horowitz M.A.* SRT Division Architectures and Implementations // *Proceedings of 13th IEEE International Symposium on Computer Arithmetic.* – 1997. – P.18-25.
7. *Ercegovac M.D., Lang T.* Division and Square-Root Algorithms: Digit-Recurrence Algorithms and Implementations. – Norwell, MA: Kluwer Academic Publishers, 1994.
8. *Oberman S.F., Flynn M.* Division algorithms and implementations // *IEEE Transactions on Computers.* – 1997. – № 8(46). – P.833–854.
9. *Tenca A.F., Ercegovac M.D.* On the Design of High-Radix On-Line Division for Long Precision // *Proceedings of the 14th IEEE Symposium on Computer Arithmetic.* – 1999. – P.44-51.
10. *Сорокин Н.Ю., Вохмин Е.В.* Высокоскоростные модули целочисленного деления на базе программируемой логики. // *Проблемы разработки перспективных микроэлектронных систем МЭС-2005: материалы Всеросс. науч.-техн. конф.* – М.: ИППМ РАН, 2005. – С.434-439.

Статья представлена к публикации членом редколлегии Чье Ен Уном.