

УДК 004.315.5

БЫСТРЫЙ ПОСЛЕДОВАТЕЛЬНЫЙ ЦЕЛОЧИСЛЕННЫЙ ДЕЛИТЕЛЬ ПО ОСНОВАНИЮ 4

А.С. Румянцев

Разработан одноктаковый блок нормализации делителя и делимого, занимающий немногим большую площадь по сравнению с последовательными (многоктаковыми) блоками нормализации. Предложен метод определения ситуации переполнения при целочисленном знаковом и беззнаковом делении, позволяющий минимизировать площадь, занимаемую аппаратными ресурсами, и снизить их энергопотребление. Приведено сравнение различных архитектурных вариантов конвейеризации устройства делителя по основанию 4, которые используют разработанный одноктаковый блок нормализации и предложенный метод определения ситуации переполнения.

Ключевые слова: целочисленное деление по основанию 4, нормализация делителя и делимого, переполнение при целочисленном знаковом и беззнаковом делении.

Введение

Целочисленное деление является одной из наиболее затратных операций в современных процессорах, так как деление обладает самым длительным временем выполнения среди всех базовых целочисленных арифметических операций [1]. Хотя операция деления встречается не так часто, как операции сложения и умножения, есть множество важных областей, которые используют эту операцию: системы рендеринга, искусственный интеллект, алгоритмы сжатия данных и т.д. [2]. Наиболее часто используется деление на основе повторов [3], так как деление на основе последовательного приближения [4] в большинстве случаев предполагает использование одноктакового умножителя или даже нескольких одноктаковых умножителей. Использование большего основания при делении на основе повторов является очевидным способом ускорения операции деления [5], но этот подход увеличивает сложность аппаратной реализации и, как следствие, приводит к увеличению занимаемой площади, энергопотреблению и соотношения цена/производительность. На сегодняшний день представлено множество подходов к реализации деления на больших основаниях [6], но некоторые аспекты реализации все еще остаются не до конца исследованными, например, эффективный по площади и энергопотреблению метод определения ситуации переполнения при делении.

В настоящей работе разработан одноктаковый блок нормализации делимого и делителя и предложен метод определения ситуации переполнения при делении для использования в каноническом устройстве делителя для 64/32-, 32/16- и 16/8-битных беззнаковых и знаковых целых чисел по основанию 4 (radix-4) [2, 3]. Разработанный блок нормализации позволяет выполнять нормализацию делимого и делителя за один такт, занимая при этом небольшую площадь в сравнении с последовательными блоками нормализации [7]. Предлагаемый метод определения ситуации переполнения при делении позволяет минимизировать площадь, занимаемую аппаратными ресурсами, и снизить их энергопотребление по сравнению с широко используемым на данный момент стандартным подходом [2, 6]. Кроме того, в работе приведено сравнение различных вариантов конвейеризации устройства делителя по основанию 4. Все рассмотренные варианты реализации устройства деления были верифицированы на корректность и синтезированы на библиотеку элементов TSMC LP120a 40 нм с использованием Synopsys DC и ICC.

Одноктаковый блок нормализации делимого и делителя

Блок нормализации является одним из самых низкоскоростных блоков в критических путях устройств деления [3]. При использовании последовательных блоков нормализации среднее время нормализации будет значительным [7], но занимаемая площадь окажется минимальной. Разработанный блок нормализации является одноктаковым, занимает небольшую площадь по сравнению с последовательным блоком нормализации и позволяет определить случай равенства делителя нулю.

Пусть x , d , q , rem обозначают делимое, делитель, частное и остаток от деления соответственно:

$$x = q * d + rem. \quad (1)$$

В формуле (1) и в последующих формулах знак «*» означает скалярное умножение, если не оговорено иное.

При реализации делителя на основе повторов с использованием таблицы выбора цифры частного необходимо, чтобы делимое и делитель были нормализованы [7]. Для нормализации произвольных делимых и делителей необходимо определить значение коэффициента нормализации ζ для получения нормализованного делителя ($d * \zeta$). Тогда, после нормализации делителя и делимого, (1) приобретет вид

$$x * \zeta = q * (d * \zeta) + rem * \zeta, |rem * \zeta| \leq |d * \zeta| - \zeta. \quad (2)$$

Основными задачами блока нормализации являются:

- нахождение позиции ведущей «1» в делителе;

- сдвиг делителя влево на определенную величину, чтобы ведущая «1» оказалась в позиции наибольшего значащего бита (2). Делимое сдвигается на соответствующую величину.

Архитектура разработанного одноклового блока нормализации с определением равенства нулю делителя показана на рис. 1 для 64-битного делимого и 32-битного делителя.

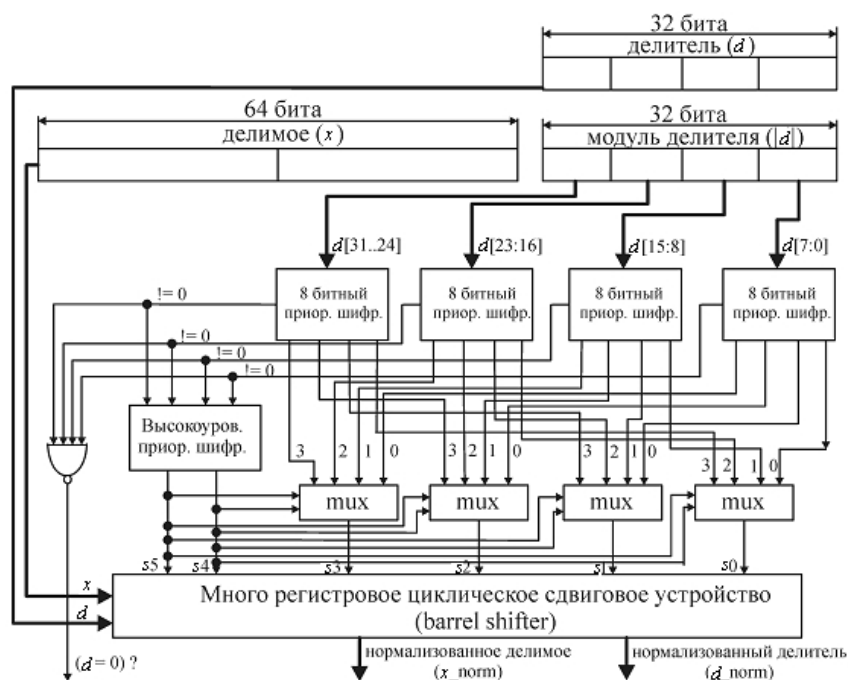


Рис. 1. Архитектура одноклового блока нормализации с определением равенства нулю делителя

Метод определения ситуации переполнения при делении

В рассматриваемом в работе делителе делимое ($2N$) больше частного (N) в два раза. При этом делитель обладает двумя режимами работы:

- беззнаковое деление: $0 < x \leq 2^{2n} - 1; 0 < d \leq 2^n - 1$;
- знаковое деление: $-2^{2n-1} \leq x \leq 2^{2n-1} - 1; -2^{n-1} \leq d \leq 2^{n-1} - 1$.

Довольно часто деление двух операндов, делимого и делителя, приводит к ситуации переполнения. В общем случае, с учетом беззнакового и знакового делений, переполнение происходит в следующих случаях.

1. Делитель d равен нулю. Заметим, что данная ситуация определяется в разработанном блоке нормализации делимого и делителя.

2. Частное q и/или остаток rem выходят за следующие диапазоны:

- при беззнаковом делении

$$0 < q \leq 2^n - 1; \tag{3}$$

$$0 < rem \leq d - 1; \tag{4}$$

- при знаковом делении

$$-2^{n-1} \leq q \leq 2^{n-1} - 1; \tag{5}$$

$$-|d| + 1 \leq rem \leq |d| - 1. \tag{6}$$

Воспользовавшись формулами (1), (3–6), можно перейти к стандартному принципу определения ситуации переполнения при делении беззнаковых и знаковых чисел [2, 6].

1. Делитель d равен нулю.

2. Не выполняются следующие условия:

- при беззнаковом делении $x < d * 2^n$

- при знаковом делении $\begin{cases} |x| < |d| * 2^{n-1}, sign(x) = sign(d); \\ |x| < |d| * 2^{n-1} + |d|, sign(x) \neq sign(d). \end{cases}$

Рассмотренный выше вариант определения ситуации переполнения при делении имеет серьезные недостатки, которые значительно увеличивают занимаемую аппаратными ресурсами площадь и критический путь устройства делителя:

- необходимость получения модуля делимого x , который в разработанном устройстве деления может достигать размера в 64 бита;
- сравнение абсолютного значения делимого по отношению к абсолютному значению сдвинутого делителя. В разработанном устройстве деления для этого потребуется использование полнофункционального 64-битного сумматора;
- при несовпадении знаков делимого и делителя необходимо будет сравнить абсолютное значение делимого и суммы абсолютного значения сдвинутого делителя с его абсолютным значением. В некоторых ситуациях для этого потребуется использование полнофункционального 32-битного сумматора.

В разработанном методе при делении используется иной подход к определению ситуации переполнения, в котором отсутствуют приведенные недостатки. Делимое x разделяется на две части Y и Z , положение которых зависит от типа деления: беззнаковое или знаковое (рис. 2).

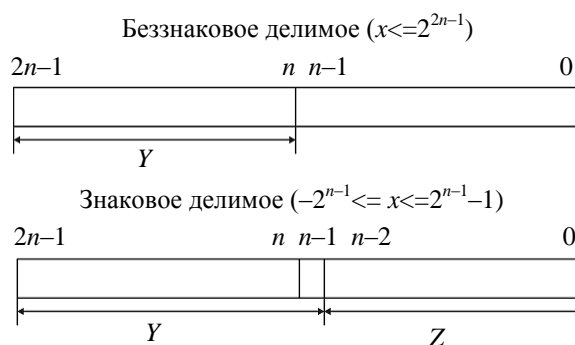


Рис. 2. Разделение делимого на две части (Y и Z)

В дальнейшем, в зависимости от знаков делимого x и делителя d , определяется ситуация переполнения:

$$1. \begin{cases} x \geq 0, d \geq 0 \rightarrow K = Y - d \\ K < 0, \text{нет переполнения} \\ K \geq 0, \text{переполнение} \end{cases}$$

$$2. \begin{cases} x < 0, d \geq 0 \rightarrow K = \text{инвертирование}(Y) - d \\ K < 0, \text{нет переполнения} \\ K = 0, \text{специальный случай. Специальный случай} : \begin{cases} Z + d > 0, \text{нет переполнения} \\ Z + d \leq 0, \text{переполнение} \end{cases} \\ K > 0, \text{переполнение} \end{cases}$$

$$3. \begin{cases} x \geq 0, d < 0 \rightarrow K = Y + d \\ K < 0, \text{нет переполнения} \\ K = 0, \text{специальный случай. Специальный случай} : \begin{cases} Z + d < 0, \text{нет переполнения} \\ Z + d \geq 0, \text{переполнение} \end{cases} \\ K > 0, \text{переполнение} \end{cases}$$

$$4. \begin{cases} x < 0, d < 0 \rightarrow K = Y - d \\ K > 0, \text{нет переполнения} \\ K = 0, \text{специальный случай. Специальный случай} : \begin{cases} Z \neq 0, \text{нет переполнения} \\ Z = 0, \text{переполнение} \end{cases} \\ K < 0, \text{переполнение} \end{cases}$$

На рис. 3 показана зависимость площади и потребляемой мощности от частоты тактирования устройства делителя на 40 нм LP120a в библиотеке элементов TSMC для стандартного метода и предложенного метода определения ситуации переполнения при делении.

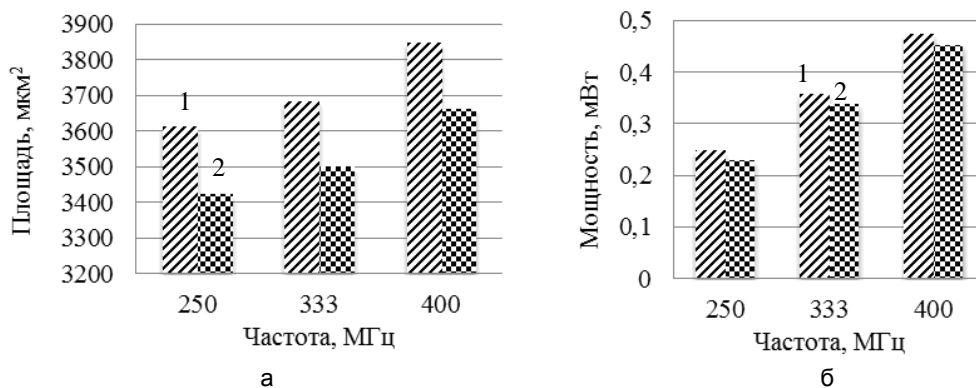


Рис. 3. Зависимость площади (а) и потребляемой мощности (б) делителя от частоты тактирования: 1 – стандартный метод, 2 – разработанный метод

Как видно из рис. 3, аппаратная реализация предложенного метода определения ситуации переполнения оказывается лучше, чем стандартный подход, занимая в среднем на 5,1% меньше площади и потребляя в среднем на 7,8% меньше мощности.

Анализ воздействия конвейеризации на устройство делителя

На рис. 4 приведена функциональная блок-схема последовательности стадий работы рассматриваемого устройства делителя.

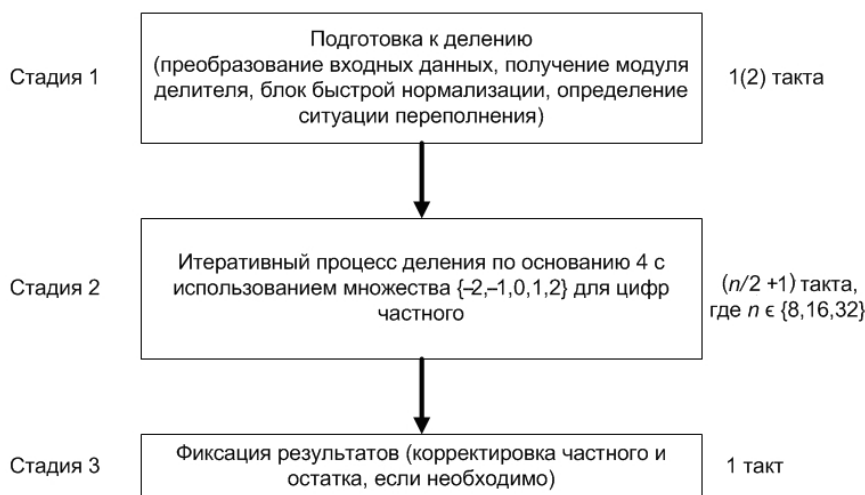


Рис. 4. Функциональная блок-схема стадий работы разработанного устройства деления

Рассмотрим различные варианты конвейеризации стадий устройства деления.

1. Стадия 1 занимает два такта, а стадия 3 – один такт.
2. Стадия 1 и стадия 3 занимают один такт.
3. Стадия 1 занимает 1 такт, а стадия 3 комбинационно объединяется со стадией 2.
4. Стадии 1 и 3 объединяются комбинационно вместе со стадией 2.

На рис. 5 показана зависимость площади и потребляемой мощности от частоты тактирования устройства делителя на 40 нм LP120a в библиотеке элементов TSMC для различных вариантов конвейеризации. Рис. 6 отображает время деления чисел разной разрядности на различных вариантах исполнения устройства деления.

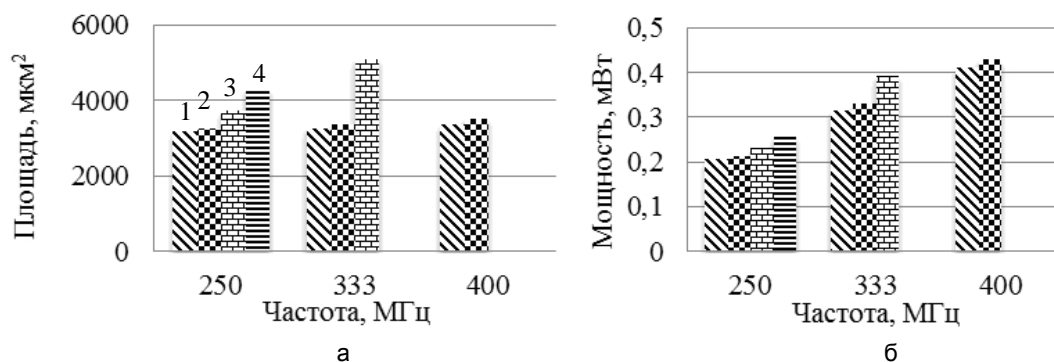


Рис. 5. Зависимость площади (а) и потребляемой мощности (б) от частоты тактирования делителя: 1 – вариант №1; 2 – вариант №2; 3 – вариант №3; 4 – вариант № 4

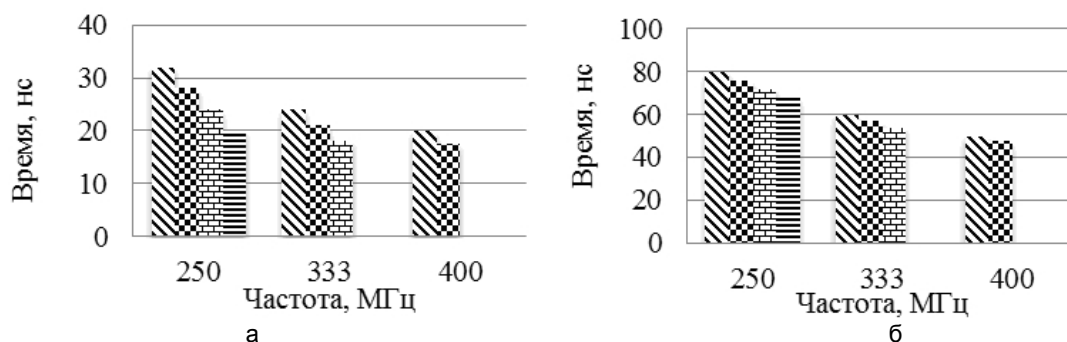


Рис. 6. Зависимость времени деления 16 на 8-битные числа (а) и 64 на 32-битные числа (б) от частоты (обозначения те же)

Из рис. 5, 6 видно, что чем более конвейеризованным становится устройство делителя, тем оно с повышением частоты занимает меньшую площадь, потребляет меньшую мощность и укладывается в ограничения целевой библиотеки элементов. Однако при этом возрастает количество тактов, необходимое для завершения операции деления. С позиций эффективности по занимаемой площади, энергопотреблению и возможности масштабирования частоты тактирования вариант № 2 конвейеризации устройства делителя выглядит более предпочтительным.

Заключение

В работе представлены новые подходы к реализации отдельных подсистем быстрого последовательного целочисленного делителя по основанию 4: одноклового блока нормализации делимого и делителя и блока определения ситуации переполнения. Разработанный блок нормализации делимого и делителя позволяет выполнять нормализацию и определение ситуации равенства нулю делителя за один такт, позволяя уменьшить время выполнения операции деления на 16% и увеличивая занимаемую площадь на 1,7% по сравнению с использованием последовательных блоков нормализации [7]. Результаты синтеза показывают, что аппаратная реализация предложенного метода определения ситуации переполнения при делении позволяет в среднем уменьшить занимаемую площадь на 5,1% и потребляемую мощность на 7,8% по сравнению с широко используемым стандартным подходом [2, 6].

Кроме того, был проведен анализ зависимости степени конвейеризации устройства делителя, использующего предложенные подходы, от занимаемой площади и потребляемой мощности. На основе произведенного сравнения можно осуществлять выбор варианта исполнения устройства деления в зависимости от производительности операции деления и планируемой частоты тактирования.

Литература

1. Intel Corp. Intel® 64 and IA-32 Architectures Software Developer's Manual // Intel Press. – 2010. – V. 1. – 540 с.
2. Omondi R. Computer Arithmetic Systems. – Prentice Hall, 1994. – 520 с.
3. Ercegovac M.D. and T. Lang. Division and square root-digit-recurrence algorithms and implementations. – Kluwer Academic Publishers, 1994. – 230 с.
4. Hwang K. Computer arithmetic: principles, architectures, and designs. – John Wiley & Sons. – 1979. – 250 с.
5. Carter T.M. and J.E. Robertson. Radix-16 signed-digit division // IEEE Trans. Computers. – 1990. – V. 12. – № 39. – С. 1424–1433.
6. Ercegovac M.D. and P. Montuschi Very high radix division with prescaling and selection by rounding // IEEE Trans. Computers. – 1994. – V. 6. – № 43. – С. 909–917.
7. Hayes J.P. Computer architecture and organization. – McGraw-Hill, 1988. – 502 с.

Румянцев Александр Сергеевич

– Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, аспирант, darkstreamray@gmail.com