УСТРОЙСТВА УМНОЖЕНИЯ ОДИНАРНОЙ ТОЧНОСТИ НА БАЗЕ FPGA

Владимир Опанасенко, Сергей Крывый, Станислав Завьялов

***Аннотация**: Определено число частичных произведений для обеспечения результата одинарной точности. В работе предложены последовательная и последовательно-параллельная структуры устройств умножения матриц. Приведены временные и аппаратурные оценки их реализации на базе FPGA. Синтезированы функциональные блоки с плавающей точкой, совместимые со стандартом IEEE–754, которые могут быть использованы в качестве библиотечного элемента при разработке сложных вычислительных устройств.*

***Ключевые слова**: одинарная точность, последовательная структура, последовательно-параллельная структура, умножение, FPGA.*

***ITHEA Keywords**: B. Hardware: B.2 ARITHMETIC AND LOGIC STRUCTURES: B.2.4 High-Speed Arithmetic*

Введение

Принцип работы произвольного устройства умножения [Майоров, 1970] можно представить совокупностью двух операций – формирования и суммирования частичных произведений, которые могут использоваться в различной последовательности. Изменение этой последовательности влияет на значение основных характеристик устройства – быстродействие и стоимость [Opanasenko, 2017]. Однако, в любом случае эти характеристики зависят от числа частичных произведение, которые необходимо сформировать для обеспечения заданной точности.

Для разрядности n сомножителей произведение в общем случае представляется $2n$ –разрядным. Удвоенная разрядность результата существенна для случая организации умножения многократной точности. В других случаях произведение ограничивают n разрядами, используя симметричное округление, так как не имеет смысла определять результат с погрешностью, намного меньшей наследственной погрешности, определяемой неточностью сомножителей (при максимуме модуля погрешности сомножителей $\varepsilon = 2^{-n-1}$ модуль погрешности произведения находится в диапазоне $0 \div 2^{-n}$).

1. Постановка задачи определения числа частичных произведений с одинарной точностью результата.

Пусть сомножители A и B представлены в виде

$$A = \sum_{i=1}^{l} a_i 2^{-ih}, \; B = \sum_{j=1}^{l} b_j 2^{-jh}, \tag{1}$$

где h - число двоичных разрядов сомножителей модуля-умножителя (МУ); $a_i \le p-1, b_j \le p-1$ - целые числа двоичного представления p –ричной цифры, $p = 2^h$; l –разрядность p –ричных сомножителей ($n = lh$). Тогда произведение определяется выражением

$$C = AB = \sum_{i=1}^{l} \sum_{j=1}^{l} a_i b_j 2^{-(i+j)h}.$$

Заметим, что частичные произведения $c_{ij} = a_i b_j$ являются $2h$ –разрядными числами, причем каждое из них представляется суммой

$$c_{ij} = d_{ij} 2^h + g_{ij}, \tag{2}$$

где $d_{ij} \le p-1$ и $g_{ij} \le p-1$ – двоичное представление старшей и младшей p –ричной цифры частичного произведения.

Частичные произведения $c_{ij}, i, j = \overline{1,l}$ упорядочим в виде ленточной матрицы шириной l таким образом, чтобы столбцы матрицы содержали компоненты, имеющие одинаковые весовые множители:

$$\left\| \begin{array}{cccccccccc} c_{1,1} & c_{1,2} & c_{1,3} & \ldots & c_{1,l} & 0 & 0 & \ldots & 0 & 0 \\ 0 & c_{2,1} & c_{2,2} & \ldots & c_{2,l-1} & c_{2,l} & 0 & \ldots & 0 & 0 \\ 0 & 0 & c_{3,1} & \ldots & c_{3,l-2} & c_{3,l-1} & c_{3,l} & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & c_{l-1,2} & c_{l-1,3} & c_{l-1,4} & \ldots & c_{l-1,l} & 0 \\ 0 & 0 & 0 & \ldots & c_{l,1} & c_{l,2} & c_{l,3} & \ldots & c_{l,l-1} & c_{l,l} \end{array} \right\| \quad (3)$$

Каждая строка содержит элементы c_{ij} с постоянным номером i, нумерация строк (сверху–вниз) соответствует возрастанию значения i, число столбцов – $(2l-1)$, сумма индексов $i+j$ для каждого столбца постоянна и возрастает при перечислении столбцов (слева-направо) от 2 до $2l$. В результате

$$C = \sum_{k=2}^{l+1} 2^{-kh} \left(\sum_{i=1}^{k-1} c_{i,\, j=k-i} + 2^{-lh} \sum_{i=k}^{l} c_{i,\, j=l+k-i} \right), \quad (4)$$

где при $i = l+1$

$$2^{-lh} \sum_{i=l+1}^{l} c_{i,\, j=l+k-i} = 0,$$

так как нижний предел суммирования больше верхнего. Чтобы разделить результат на старшие и младшие разряды (по l p –ричных разрядов), преобразуем (4) к виду

$$C = \sum_{k=2}^{l} 2^{-kh} \sum_{i=1}^{k-1} c_{i,\, j=k-i} + 2^{-(l+1)h} \sum_{i=1}^{l} c_{i,\, j=l+1-i} + 2^{-(l+2)h} \sum_{i=2}^{l} c_{i,\, j=l+2-i} + \sum_{k=3}^{l} 2^{-(l+k)h} \sum_{i=k}^{l} c_{i,\, j=l+k-i}, \quad (5)$$

Тогда с учетом (2) получим

$$C=\sum_{k=2}^{l}2^{-kh}\sum_{i=1}^{k-1}c_{i,j=k-i}+2^{-lh}\sum_{i=1}^{l}d_{i,j=l+1-i}+2^{-(l+1)h}\sum_{i=1}^{l}g_{i,j=l+1-i}+2^{-(l+1)h}\sum_{i=2}^{l}d_{i,j=l+2-i}+$$
$$+2^{-(l+2)h}\sum_{i=2}^{l}g_{i,j=l+2-i}+2^{-(l+3)h}\sum_{i=3}^{l}d_{i,j=l+3-i}+2^{-(l+3)h}\sum_{i=3}^{l}g_{i,j=l+3-i}+\sum_{k=4}^{l}2^{-lh}\sum_{i=k}^{l}c_{i,j=l+k-i}, \quad (6)$$

Первые два слагаемых в (6) составляют основу l старших p –ричных разрядов произведения, остальные члены – основу младших разрядов. Если в (6) принять $g_{i,j=l+1-i}\approx p=2^h$, $i=\overline{1,l}$ а $d_{i,j=l+2-i}\approx p=2^h$, $i=\overline{2,l}$, то

$$2^{-(l+1)h}\sum_{i=1}^{l}g_{i,j=l+1-i}+2^{-(l+1)h}\sum_{i=2}^{l}d_{i,j=l+2-i}<(2l-1)2^{-lh}, \quad (7)$$

$$2^{-(l+2)h}\sum_{i=2}^{l}g_{i,j=l+1-i}<(l-1)2^{-(l+1)h},\ \ 2^{-(l+2)h}\sum_{i=3}^{l}d_{i,j=l+3-i}<(l-3)2^{-(l+1)h}. \quad (8)$$

Из оценок (7) и (8) следует, что для получения произведения однократной длины n (совпадает с разрядностью сомножителей) с точностью до единицы или половины младшего двоичного разряда 2^{-lh} достаточно ограничиться первыми тремя слагаемыми (5), так как старшие разряды исключенного слагаемого $\sum_{k=3}^{l}2^{-(l+k)h}\sum_{i=k}^{k-1}c_{i,j=l+k-i}$ согласно (8) вносят погрешность порядка $(l-3)\times 2^{-(l+1)h}=2^{-(l+1)h+\log_2(l-3)}$.

При достаточно большом h и реальных значениях $l, h>\log_2(l-3)$, например, при $h=8$ и $l=8$ (соответствует 64-разрядным двоичным числам) погрешность за счет отбрасывания младших частичных произведений не превышает 2^{-69}, что существенно меньше 2^{-65} (погрешность произведения однократной длины при симметричном округлении).

Такой подход соответствует вычислению частичных произведений первых $(l+1)$ столбцов матрицы (3) с весовыми множителями 2^{-2h} до $2^{-(l+2)h}$ и при сохранении заданной точности обеспечивает существенную экономию вычислительных ресурсов. При

использовании МУ для получения произведения однократной длины l –разрядных p –ричных сомножителей вместо вычисления в общем случае l^2 частичных произведений достаточно выполнить $1/2(l^2+3l-2)$ перемножений, что приводит к экономии времени и числа МУ, пропорционально разности

$$l^2-\frac{l^2+3l-2}{2}=\frac{l^2+3l+2}{2}.$$

2. Распределенная реконфигурируемая обработка на примере перемножения матриц

Одной из основных особенностей программируемой логики является возможность использования принципа параллельной обработки данных при решении широкого круга задач. Увеличение ресурсов современной программируемой логики и уменьшение их стоимости, в частности ПЛИС типа FPGA фирмы Xilinx, позволяют существенно повысить быстродействие разрабатываемых устройств и реализовать аппаратно алгоритмы, работающие в реальном режиме времени. Распараллеливание вычислений или логических операций может осуществляться как на уровне разрядов представления информации, так и на уровне блоков, выполняющих соответствующие алгоритмы математической модели.

Рассмотрим реализацию алгоритма перемножения матрицы $A=\|a_{ij}\|$ ($\forall i=1\div n, j=1\div m$) на матрицу $B=\|b_{jk}\|$ ($\forall k=1\div r$). Результирующая матрица $C=\|c_{ik}\|$ размером $n\times r$ формируется следующим образом:

$$C=AB=\|a_{ij}\|\times\|b_{jk}\|=\|c_{ik}\|,$$

где

$$c_{ik}=\sum_{i=1}^{n}a_{ij}b_{jk}. \qquad (9)$$

Таким образом, согласно (9), каждый j –й элемент i –ой строки матрицы A последовательно умножается на соответствующий j –й элемент столбца матрицы B и

полученные частичные произведения суммируются, формируя элемент c_{ik} матрицы $C = \| c_{ik} \|$.

Для определения каждого элемента результирующей матрицы используются операции умножения и суммирования частичных произведений. Суммирование может осуществляться двумя способами: накоплением (аккумулированием) частичных произведений при последовательном их поступлении на вход аккумулятора с выхода МУ и параллельном суммировании частичных произведений. Первый способ предполагает наличие блока, выполняющего умножение и суммирование (накопление) полученных частичных произведений. Второй способ использует набор умножителей и многовходовый сумматор для получения элемента результирующей матрицы.

Эти способы реализованы следующими вариантами:

– последовательный (ПС), когда обрабатывающее поле состоит из одного блока, последовательно вычисляющего сумму парных произведений в (9), структурная организация представлена на рис. 1;

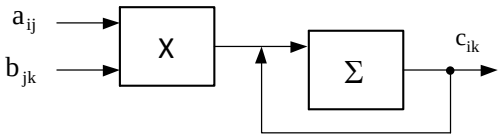

Рис. 1. Структура ПС перемножения матриц

– параллельно–последовательный (ПП1), когда обрабатывающее поле содержит множество блоков (рис. 2), число которых соответствуют количеству ($i = n$) строк матрицы A, с помощью которых одновременно вычисляется k–й ($k = 1 \div r$) элемент c_{ik} столбца и далее последовательно формируются все столбцы матрицы $C = \| c_{ik} \|$ в (9);

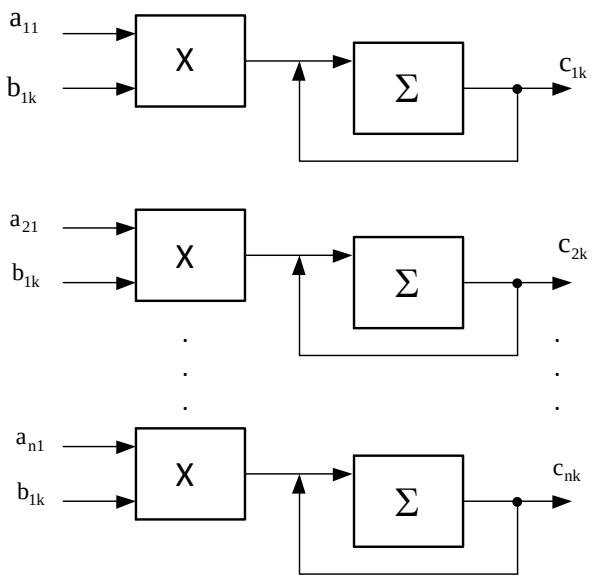

Рис. 2. Структура ПП1 перемножения матриц

– параллельно–последовательный (ПП2), когда обрабатывающее поле содержит такое количество блоков (рис. 3), в котором число МУ соответствуют количеству ($i = n$) строк матрицы A, параллельно реализуя, таким образом, вычисление одного элемента c_{ik} , а далее последовательно вычисляются остальные элементы c_{ik} матрицы $C = \| c_{ik} \|$.

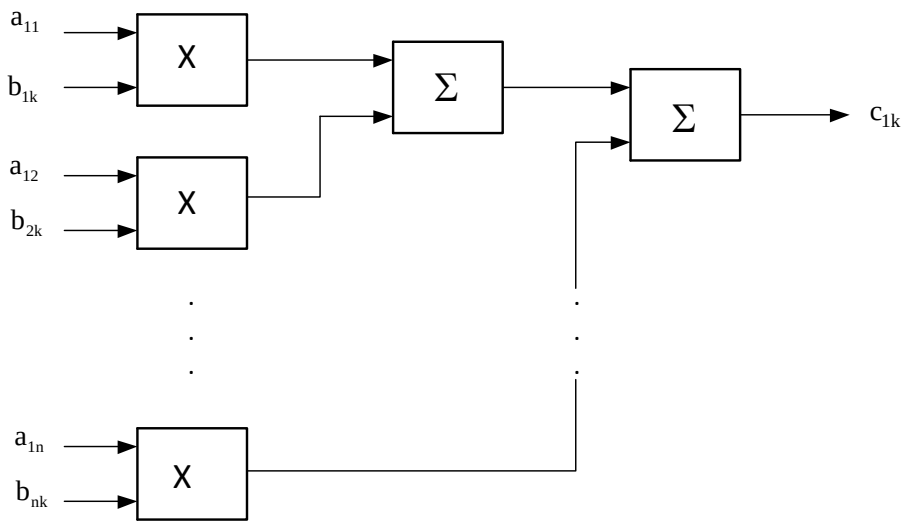

Рис. 3. Структура ПП2 перемножения матриц

Распределенная арифметика реализует арифметические операции и основана на использовании логических функциональных генераторов (LUT). Поскольку базовая логическая ячейка ПЛИС типа FPGA фирмы Xilinx содержит LUT, то архитектура FPGA позволяет на их основе реализовать как логические, так и арифметические функции, в том числе умножение, широко используемое в цифровой обработке сигналов.

Рассмотрим реализацию устройства, выполняющего умножение квадратных матриц порядка $m = 10$ для целых 16–разрядных чисел, реализованного в кристалле серии Virtex со степенью быстродействия – 8. Аппаратные затраты на реализацию определяются логической емкостью кристалла, т.е. соответствующим количеством слайсов. Количество затраченных слайсов включает в себя входные, выходные и промежуточные регистры для реализации конвейерного способа вычислений. Время выполнения операции умножения двух 16–разрядных чисел с накоплением 32–разрядной суммы (суммированием результата умножения с числом, находящимся в аккумуляторе) для указанного типа кристалла составляет 6,424 нс. В табл. 1 приведены аппаратные и временные оценки для рассмотренных вариантов реализации.

Таблица 1.

Вариант реализации алгоритма умножения матриц	Количество умножителей сумматоров	Быстродействие (полное время перемножения матриц), нс	Аппаратные затраты (количество слайсов)
ПС	1/1	6424	181
ПП1	10/10	642,4	1810
ПП2	10/1	890	1665

3. Представление чисел в формате с плавающей точкой

Формат представления чисел с плавающей точкой использует своего рода "подвижное окно" точности, соответствующее масштабу числа. В стандарте IEEE 754 [Hollash, 2018] формат одинарной точности представлен 32 битами – 1 бит для знака, 8 бит для порядка и 23 бита для дроби мантиссы. Однако данный формат предполагает наличие «скрытого» старшего бита (f_0), так что мантисса на самом деле представлена 24 битами ($p = 24$).

Представим число с плавающей точкой в следующем виде: $A=(-1)^{sign}\times s^{e}\times F$, где: sign – знак числа; s – основание системы счисления (в данном случае $s=2$); e – порядок (значение которого для формата одинарной точности представлено со смещением на $b=+127$); $F=f_0+f$ – мантисса; $f=\left(f_1s^{-1}+f_2s^{-2}+...f_is^{-i}+...+f_{p-1}s^{-(p-1)}\right)$, $(0\le f_i<s)$ – дробь мантиссы; p – разрядность мантиссы ($p=24$).

Если старшая значащая цифра отлична от нуля ($f_0\neq 0$), то число считается нормализованным – $A=(-1)^{sign}\times 2^{(e-b)}\times 1.f$. Нормализованные числа представляются диапазоном от $\pm 2^{-126}$ до $(2-2^{-23})\times 2^{127}$.

Самые большие и самые малые допустимые величины порядков принимают значения $e_{\max}=+127$ и $e_{\min}=-126$. Так как имеется s^{p} возможных значений мантиссы, и ($e_{\max}-e_{\min}+1$) возможных значений показателей, то число с плавающей запятой может быть закодировано следующим количеством битов -$\left[\log_2(e_{\max}-e_{\min}+1)\right]+[\log_2(s^{p})]+1$, где последнее слагаемое (+1) предназначено для знакового разряда.

Значение знакового разряда «0» соответствует положительному числу, а «1» – отрицательному.

Поле порядка должно представлять положительные и отрицательные значения порядка. К фактическому порядку добавляется смещение – для формата одинарной точности это значение равно 127. Таким образом, порядок нулевого значения предполагает, что в поле порядка сохранено значение 127. Например, сохраненное значение порядка 200 указывает порядок (200 – 127) или фактическое значение 73. Значения порядка –127 (все «0») и + 128 (все «1») зарезервированы для специальных чисел.

4. Core–блоки функциональных устройств с плавающей точкой

Рассмотрим разработку устройств, выполняющих операции с плавающей точкой (алгебраическое сложение, умножение, деление, сравнение и преобразование форматов) над 32–разрядными операндами в соответствии со стандартом IEEE–754 [Hollash, 2018]. Обобщенная структура функциональных блоков с плавающей точкой представлена на рис. 4 и состоит из трех составных модулей: модуль контроля входных аргументов (МКА); функциональный модуль (ФМ) и модуль формирования результата (МФР). Описание

модулей выполнено на VHDL–языке, при разработке использован синтезатор FPGA Compiler II фирмы Synopsys, для формирования Core–блоков применена система CORE Generator System фирмы Xilinx. Разработанные модули верифицированы методом моделирования с определением временных и аппаратных параметров. Модули представляют собой законченные типовые технические решения и могут быть использованы в других проектах в качестве soft cores.

МКА преобразует входные данные, анализирует их на соответствие стандарту IEEE–754 с формированием соответствующих признаков. Соответствующие числа и информацию относительно классов входных данных выдает как результаты функциональному модулю с плавающей точкой.

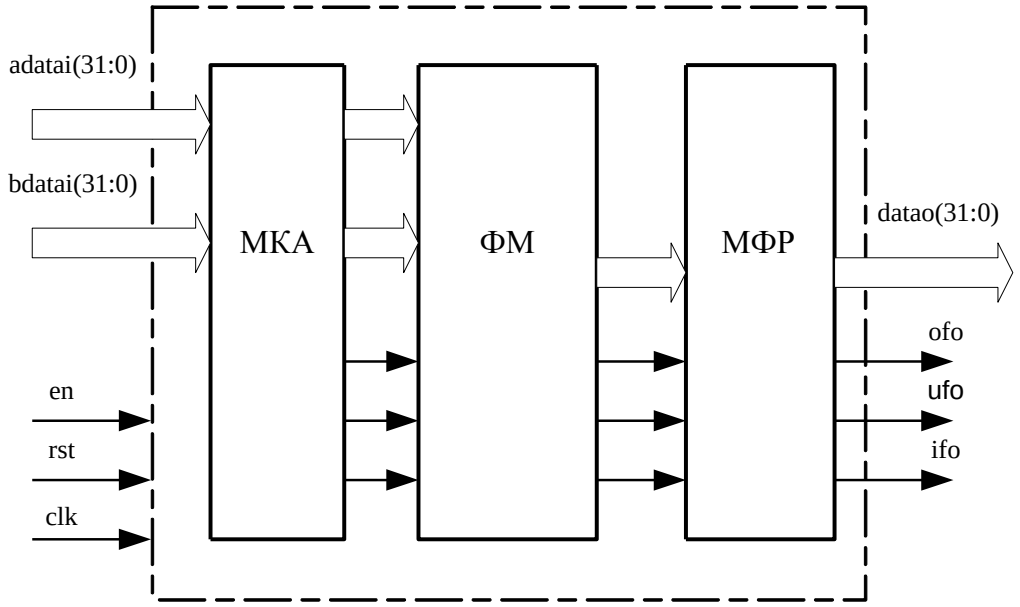

Рис. 4. Структура функционального блока с плавающей точкой

ФМ выполняет заданную операцию с плавающей запятой с формированием соответствующих признаков. МФР выполняет согласование формата результирующих данных со стандартом IEEE–754 и окончательную установку флагов. Входы и выходы блока с плавающей точкой не привязаны к фиксированным контактам ввода-вывода конкретного кристалла ПЛИС, поэтому допускается использование произвольного кристалла. Назначение входов и выходов блока показано на рис. 4: clk – глобальный сигнал clock; rst – глобальный сигнал reset; en – сигнал Enable; adatai (31:0) – входная шина

данных A; bdatai (31:0) – входная шина данных B; datao (31:0) – выходная шина данных; ofo – флаг “Overflow”; ufo – флаг “Underflow”; ifo – флаг «Недопустимая операция».

Для согласования полученного результата преобразования со стандартом IEEE–754 необходимо представить числа в нормализованном виде. Поэтому требуется определить месторасположение старшей значащей «1» и выполнить сдвиг в сторону старших значащих бит на требуемое количество разрядов с одновременным вычитанием этого значения из результирующего порядка. Наличие мощных логических ресурсов в кристаллах серии Virtex позволяет эту процедуру ускорить путем быстрого определения количества сдвигов. Тогда, в отличие от реализации последовательного сдвига с одновременным анализом старшего значащего бита, выполняется параллельный сдвиг на требуемое количество разрядов для нормализации мантиссы.

Суммирование с плавающей точкой включает пять строго последовательных операций: сравнение порядков, сдвиг вправо мантиссы меньшего числа, суммирование мантисс, поиск левой единицы мантиссы результата, нормализация мантиссы результата.

Для реализации операции поиска левой единицы предлагается использование приоритетного шифратора. Пусть имеется ($n = 24$) значащих бит мантиссы. Требуется определить номер старшего «ненулевого» разряда и выполнить нормализацию мантиссы $F = \{f_{23}, f_{22}, ..., f_i, ..., f_0\}$.

Приоритетный шифратор представляет собой комбинационную схему, имеющую n входов и ($Ent\{\log_2 n\}$) выходов, которая состоит из двух последовательно соединенных схем - первая выделяет старшую значащую единицу, а вторая ее номер (количество требуемых сдвигов) в операнде.

Первая схема имеет n входов и n выходов, реализуя следующую систему логических уравнений:

$$a_i = f_i \left(\bigcap_{i=i+1}^{(n-1)} (\overline{f_i}) \right), \ \forall i = 0 \div (n-1). \qquad (5)$$

Вторая схема имеет n входов и ($Ent\{\log_2 n\}$) выходов, реализуя следующую систему логических уравнений:

$$y_j = \bigcup_{k=1}^{N=Ent\{(n-1)/(2^{(j+1)})\}} \left[\bigcup_{i=2^j+(k-1)\times 2^{(j+1)}}^{2^j+(2^j-1)+(k-1)\times 2^{(j+1)}} a_i \right], \ (\forall j = 0 \div (Ent\{\log_2 n\} - 1)). \qquad (6)$$

Таким образом, выражения (5) и (6) позволяют синтезировать приоритетный шифратор произвольной разрядности, представляющий собой параметрический модуль, который может быть использован при разработке новых проектов другими пользователями.

Умножение с плавающей точкой включает два основных действия – вычисление произведения мантисс и нормализацию результирующей мантиссы. Результирующий порядок (как сумма порядков сомножителей) вычисляется параллельно.

При разработке типовых модулей, так же как и при разработке обычных проектов, целесообразно использование уже хорошо отработанных доступных IP–Core.

Рассмотрим пример построения 32–разрядного блока умножения с плавающей точкой. Блок состоит из трех элементов, первые два из которых, в соответствии с рис. 4, входят в функциональный модуль ФМ, а третий – в функциональный модуль МФР [Palagin, 2017].

Первый элемент формирует 24-разрядные операнды для блока умножения (“1” в старшем – 23–м разряде и 23 разряда дроби мантиссы), суммирует порядки перемножаемых чисел (8 разрядов) и определяет знак результата. Второй элемент выполняет операцию умножения и формируется средствами Core–генератора фирмы Xilinx.

Третий элемент выполняет проверку условий и формирование результата. Проверяются следующие условия: если сумма порядков чисел равна или более 255, формируется сигнал переполнения (overflow); если 24 разряд произведения равен “1”, то производится сдвиг произведения на один разряд в сторону младших разрядов и увеличение порядка на единицу; если, после увеличения порядка на единицу, значение порядок становится равным 255, то формируется сигнал переполнения.

При использовании элемента умножения комбинационного типа результат умножения формируется на такте, следующем за тактом регистрации операндов. В тех случаях, когда приходится перемножать массивы чисел, поступающих синхронно с какой-либо тактовой последовательностью CLK, предпочтительно использование элемента умножения конвейерного типа. В разработанном модуле использованы элементы умножения, как комбинационного типа, так и с 6–уровневым (при реализации на LUT) или 2–уровневым (при использовании встроенных блоков умножения разрядности 18x18 в кристаллах Virtex) конвейером, что позволяет за счет увеличения тактовой частоты существенно уменьшить

время перемножения массивов чисел. Для временного согласования в первый элемент модуля в этом случае введены четыре или два последовательно включаемых регистра для конвейерной передачи на выход порядка и знака произведения. Задержка (Latency) между регистрацией первых операндов и регистрацией первого произведения модуля умножения равна 5 или 3 периодам CLK соответственно при использовании 4–уровневого или 2–уровневого конвейерного элемента умножения.

На рис. 5 изображена диаграмма работы модуля контроля МКА, выполненная средствами редактора State Editor.

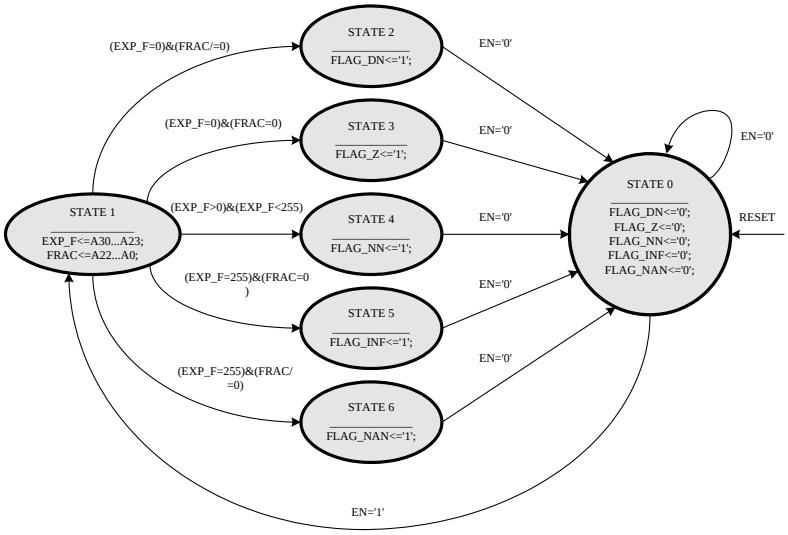

Рис. 5. Диаграмма работы модуля МКА, выполненного средствами редактора State Editor

На первом такте при наличии сигнала EN=1 блок переходит в состояние STATE1, на котором из входного операнда A (разряды 0–31) формируются сигналы EXP_F (порядок числа – разряды 23–30) и FRAC (дробь мантиссы – разряды 0–22).

Далее производится проверка условий, при выполнении одного из которых блок переходит на втором такте в одно из состояний (STATE2 – STATE6) с формированием соответствующего флага:

– если значение порядка равно нулю, а дробь мантиссы ненулевая, то входной операнд является ненормализованным числом;

– если значения порядка и дроби мантиссы равны нулю, то входной операнд является нулем;

– если значение порядка больше нуля и меньше 255, то входной операнд является нормализованным числом;

– если значение порядка равно 255 и дробь мантиссы нулевая, то входной операнд является бесконечностью ($\pm\infty$);

– если значения порядка равно 255 и дробь мантиссы ненулевая, то входной операнд не является вещественным числом (NAN).

Переход блока в исходное состояние производится по сигналу сброса (RESET) или установке сигнала EN в нулевое состояние.

В кристалле серии Spartan–II (XC2S50–5) блок занимает 46 слайсов (Slices) и работает с тактовой частотой 103 МГц.

Преимущество предложенных реализаций по сравнению с известными достигается за счет оптимального распределения описаний составляющих модулей в разных режимах, а также оригинального приоритетного шифратора, который позволяет определить номер старшей значащей «единицы» для последующего выполнения операции нормализации мантиссы за один временной такт.

В табл. 2 представлены сравнительные оценки аппаратных ресурсов и производительности разработанных модулей умножения, реализованных с использованием Core фирмы Xilinx, с аналогичными модулями фирмы Digital Core Design.

Таблица 2. Сравнительные оценки по аппаратным ресурсам и производительности

Серия и тип кристалла	Модуль фирмы Digital Core Design		Модуль с использованием Core фирмы Xilinx			
	Ресурсы (кол-во Slices)	Частота, МГц	Без конвейера		С конвейером	
			Ресурсы (кол-во Slices)	Частота, МГц	Ресурсы (кол-во Slices)	Частота, МГц
Spartan-II 2S200-6	970	47	382	42	416	75
Virtex V300-6	963	45	385	38	425	90
Virtex-II (Multiplier 18x18) 2V250-5	677+4 Multiplier 18x18	74	110+4 Multiplier 18x18	52	170+4 Multiplier 18x18	83
Virtex-II (Multiplier LUT) 2V250-5	-	-	386	49	427	128

Ресурсы оцениваются количеством слайсов (Slices), каждый из которых содержит 2 логические ячейки (Logic Cell), состоящих из функционального генератора с четырьмя входами, логики ускоренного переноса и запоминающего элемента (триггера). Производительность оценивается частотой синхропоследовательности CLK.

Аппаратные ресурсы оценены относительно известных реализаций: $\Delta Q_i = Q_0 / Q_\mu$, где: Q_0 – аппаратные затраты модуля фирмы Digital Core Design; Q_μ – аппаратные затраты предложенного модуля; Q_1 – аппаратные затраты модуля без конвейера; Q_2 – аппаратные затраты модуля с конвейерной организацией.

Для кристалла ПЛИС типа 2S200–6: $\Delta T_1 = T_0 / T_1 = 2,54$; $\Delta T_2 = T_0 / T_2 = 2,33$.

Для кристалла ПЛИС типа V300–6: $\Delta T_1 = T_0 / T_1 = 2,5$; $\Delta T_2 = T_0 / T_2 = 2,27$.

Для кристалла ПЛИС типа 2V250–5: $\Delta T_1 = T_0 / T_1 = 6,15$; $\Delta T_2 = T_0 / T_2 = 9,67$.

Вариант реализации модуля на кристалле 2V250-5 с использованием LUT отсутствует в предложениях фирмы Digital Core Design, однако по аппаратным затратам он сравним с предложенной реализаций на кристалле V300–6, но позволяет работать (примерно на треть) с большей тактовой частотой.

Заключение

Полученное аналитическим путем число частичных произведений обеспечивает результат одинарной точности. В работе предложены последовательная и последовательно-параллельная структуры устройств перемножения матриц. Приведен пример для таких устройств перемножения матриц с временными и аппаратурными оценками их реализации.

Синтезированные функциональные блоки с плавающей точкой, совместимые со стандартом IEEE–754, могут быть использованы в качестве библиотечного элемента при разработке сложных вычислительных устройств.

Библиография

[Майоров, 1970] С.А. Майоров, Г.И. Новиков. Структура цифровых вычислительных машин. Ленинград: Машиностроение, 1970. 480 с.

[Палагин, 2006] Палагин А.В., Опанасенко В.Н. Реконфигурируемые вычислительные системы. Киев: Просвіта, 2006. 295 с.

[Opanasenko, 2017] Opanasenko V.N., Kryvyi S.L. Synthesis of neural-like networks on the basis of conversion of cyclic Hamming codes. *Cybernetics and Systems Analysis*. 2017. Vol. 53, N.4. P. 627–635. DOI: DOI 10.1007/s10559-017-9965-z.

[Hollash, 2018] Hollash S. IEEE Standard 754 Floating Point Numbers. Available at https://steve.hollasch.net/cgindex/coding/ieeefloat.html.

[Palagin, 2017] Palagin A., Opanasenko V. The implementation of extended arithmetic's on FPGA-based structures. Proceedings of the 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS'2017. (21–23 September 2017, Bucharest, Romania). 2017. Vol. 2. P. 1014–1019. DOI: DOI 10.1109/IDAACS.2017.8095239.

Сведения об авторах

Опанасенко Владимир Николаевич *– профессор, доктор технических наук, ведущий научный сотрудник Института кибернетики им. В.М. Глушкова НАН Украины, Украина, Киев, 03187, просп. Глушкова, 40;* ***e-mail****: opanasenkovm@nas.gov.ua*

Крывый Сергей Лукьянович *– профессор, доктор физико-математических наук, профессор Киевского национального университета им. Тараса Шевченко, Украина, Киев, 03187, просп. Глушкова, 4д, Факультет кибернетики;* ***e-mail:*** *krivoi@i.com.ua*

Завьялов Станислав Борисович *– кандидат технических наук, директор ООО «Радионикс», Украина, Киев;* ***e-mail:*** *radionix13@gmail.com.*

FPGA-based single accuracy multiplication devices

Volodymyr Opanasenko, Sergii Kryvyi, Stanislaw Zavyalov

***Abstract**: The number of partial products is determined to ensure a single precision result. The paper proposes a serial and serial-parallel structure of matrix multiplication devices. Time and hardware assessments of their implementation on the basis of FPGA are given. Functional floating point blocks are synthesized that are compatible with the IEEE-754 standard, which can be used as a library element in the development of complex computing devices.*

***Keywords**: single precision, serial structure, serial-parallel structure, multiplication, FPGA.*