УДК 681.3

А.К. ТУМАНЯН, А.Г. ОГАНЕСЯН, А.А. АМАЗАСПЯН, А.Г. КАМАЛЯН

СИНТЕЗ УСТРОЙСТВА ИЗВЛЕЧЕНИЯ КОРНЯ КВАДРАТНОГО НА ОСНОВЕ FPGA

Рассматриваются вопросы проектирования устройства извлечения квадратного корня из целого числа и из числа с плавающей запятой. Произведено сравнение двух реализаций устройства - комбинационного и последовательного типа (с использованием FSM), полученных в результате синтеза с применением системы проектирования XILINX ISE Design Suite 14.2.

Ключевые слова: FPGA, Verilog, LUT, CLB, квадратный корень.

Операция извлечения корня квадратного из целого числа. В статье рассматривается проектирование устройства извлечения корня квадратного с применением системы проектирования ISE Design Suite 14.2. В последнее время FPGA (ПЛИС) находят широкое применение, так как их характеристики постоянно улучшаются, а объем выпуска увеличивается.

Области применения FPGA:

1. Построение реконфигурируемых систем.
2. Окончательная продукция при изготовлении малых партий изделий.
3. Отладка прототипов при проектировании (отработка проектов).
4. Проектирование встроенных систем.

В настоящее время реконфигурируемые системы, настраиваемые на исполнение алгоритмов, широко используются для построения высокопроизводительных систем [2]. Программируемая логика позволяет подойти к выпуску микроэлектронной продукции не только конструкторским организациям, но и небольшим лабораториям.

Первые FPGA содержали конфигурируемые логические блоки (CLB), в состав которых входили логический генератор (LUT –Look-up Table) на базе статической памяти (SRAM), программируемые мультиплексоры и триггеры. Современные FPGA содержат в своем составе умножители, которые в дальнейшем трансформировались в блоки цифровой обработки сигналов, встроенную память (Block RAM), блоки ввода-вывода, управления синхронизацией, интерфейсные блоки. Роль основного логического элемента играет LUT, представляющая собой,

как правило, однобитное запоминающее устройство. CLB связываются между собой системой программируемых соединений.

Основной целью статьи является задача извлечения квадратного корня из целого и дробного числа и синтез соответствующего устройства на FPGA.

Описание алгоритма извлечения корня из двоичного целого числа. Из целого числа можно извлечь или точный корень, или приближенный с точностью до 1. Если из данного целого числа не извлекается точный целый корень (получается при извлечении остаток), то из такого числа нельзя найти и дробный точный корень, так как всякая дробь, не равная целому числу, будучи умножена сама на себя, дает в произведении тоже дробь, а не целое число [1].

Для квадратов чисел верно следующее выражение:

$1+3=2^2$; $1+3+5=3^2$; $1+3+5+7+9+11+13+15+17=9^2$.

$\Sigma(2k-1)=N^2$, где k изменяется от 1 до N.

Простейший алгоритм вычисления квадратного корня заключается в вычитании из исходного числа всех нечетных чисел, начиная с 1, в порядке возрастания до тех пор, пока остаток не станет меньше очередного нечетного числа. Этот метод требует большого времени вычислений, число шагов (вычитаний) равно N.

Полезно иметь начальное значение, начиная с которого можно будет продолжать вычитания. Обозначим число цифр в двоичном числе через d. Начальное значение числа, с которого можно начать вычитание, $A=2^{\lfloor D/2 \rfloor}$. Получить это число можно путем сдвига числа 1 на $\lfloor d/2 \rfloor$ разрядов влево. Затем увеличим это число на 1, начнем вычитания (в каждом следующем шаге увеличиваем вычитаемое на 2).

Число шагов алгоритма также будет очень большим – примерно N/2 (где N^2 – ближайший точный квадрат, не превышающий исходное число).

Рассмотрим более быстродействующий алгоритм извлечения корня квадратного.

Первый способ. Рассмотрим пример извлечения квадратного корня из двоичного числа a = 10 01 11 10. Извлечение корня из двоичного числа очень мало отличается от извлечения корня из десятичного числа.

Разбиваем число на группы по два разряда. Корень из первой группы может быть только 0 или 1. Извлекаем корень из первой слева группы (10) исходного числа. Первая цифра результата равна 1. Вычитаем 1 из этой пары цифр. Удваиваем промежуточный результат после первого шага и добавляем справа 1, предполагая, что очередная цифра равна 1. Добавляем два разряда к полученной в первом шаге разности и вычитаем 101. Если разность положительна или равна 0, то следующая цифра результата равна 1. При получении отрицательной разности очередная цифра принимается равной 0. Остаток восстанавливается, к нему снова добавляется следующая пара цифр исходного числа. Теперь вычитается 1101. Остаток отрицательный, следовательно, третья цифра слева не 1, а 0.

Пример: a= _10 01 11 10 c = 1100; Остаток r = 1110;
c_3=1;1×1=1; 01
101; _ 01 01
c_2=1; 01 01
1101; _ 00 0011
1101
c_1=1; _ 001110
11001; 11001

Последняя цифра также равна нулю (c_0=0), т.к. остаток отрицательный. Эта процедура выполняется n/2 раз (4 раза в данном примере).

Результат будет получен за n/2 шагов. В каждом шаге алгоритма, приписывая исходному числу по две цифры, получаем одну цифру результата.

Второй способ. Рассмотрим извлечение квадратного корня на примере: a = 00 01 11 10 01.

Первая цифра результата будет равна 0 . Затем на каждом шаге производим следующее:

- предполагаем, что следующей цифрой будет 1;
- определяем квадрат полученного частичного результата (y_i^2);
- если y_i^2<=a_i, то очередная цифра результата остается равной 1, иначе заменяем ее нулем.

В рассматриваемом примере в первом такте ai = 00, во втором такте - 0001, в следующем – 00 01 11, в последнем 5-м такте – 00 01 11 10 01.

Извлечение корня представлено в табл.1.

Таблица 1

a_i	y_i0	y_i^2	y_i1 (result)
00	0		
00 01=00 01	01	0001	01
00 01 11	011	00 10 01	010
00 10 01>00 01 10			
00 01 11 10	0101	00011001	0101
00011001<00 01 11 10			
00 01 11 10 01	01011	0001111001	01011
00 01 11 10 01=0001111001			

y_i0 – предполагаемое значение частичного результата на i-ом шаге;

yi1 – уточненное значение частичного результата после i-го шага.

По существу, это тот же способ, вычитание здесь заменено умножением [3].

Для устройства комбинационного типа, реализующего данный алгоритм, составлено data flow описание на Verilog. Число схем сравнения и умножителей равно m-1, где m - число разрядов результата. Разрядность соответствующих устройств растет от 2-х до m.

Эта реализация наиболее удобна для FPGA, так как они содержат много комбинационной логики (например, число LUT в FPGA Spartan-3E (XC3S100E) равно 1920, а в FPGA Virtex-7 число LUT выше 55200) . Операция выполняется за один такт.

Сравним данную схему с реализацией в виде последовательного устройства.

При описании функционального устройства последовательного типа с FSM требуется разработка структурной схемы устройства (рис.1).

Операция выполняется за n/2 тактов. Перед началом операции в регистр маски (r_mask) записывается код 11 00 00 ....00, а в регистр rb1 – код 100…0**.** Разряды r_mask[n-1:n-2] выбирают старшую пару анализируемых разрядов.

После выполнения очередного такта r_mask сдвигается вправо на два разряда. При этом старшие единицы сохраняются. В последнем (n/2-1)-м такте в r_mask будут все единицы.

В rb1 содержится унитарный код. В каждом такте ”1” сдвигается вправо на один бит (в i-ом такте “1” будет находиться в i-ом, считая слева, разряде).

В регистре rb в начале i-го такта находится промежуточный результат (значение корня) после i-1 –го такта. Если предположение, что i-я цифра результата равна 1, не подтверждается, то i-й разряд устанавливается в 0, иначе - в “1” .

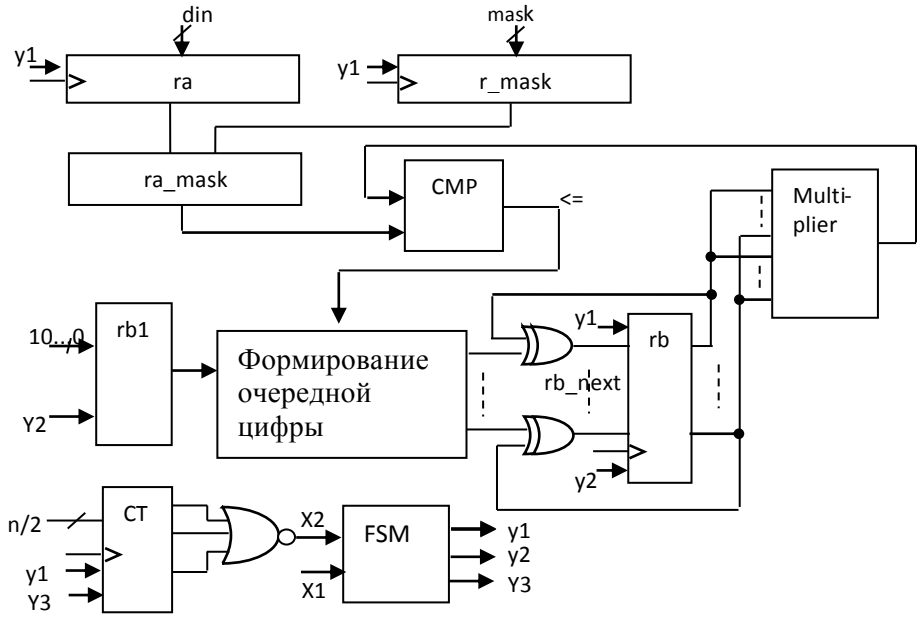


Рис. 1. Структурная схема устройства извлечения корня квадратного

Выходы rb подаются на комбинационный умножитель, на выходе которого появляется $(rb)^2$. Код на выходе ra_mask сравнивается с rb^2. Если rb^2 <= ra_mask,

то очередная цифра результата будет равна 1, иначе – 0. Выбор осуществляется мультиплексором.

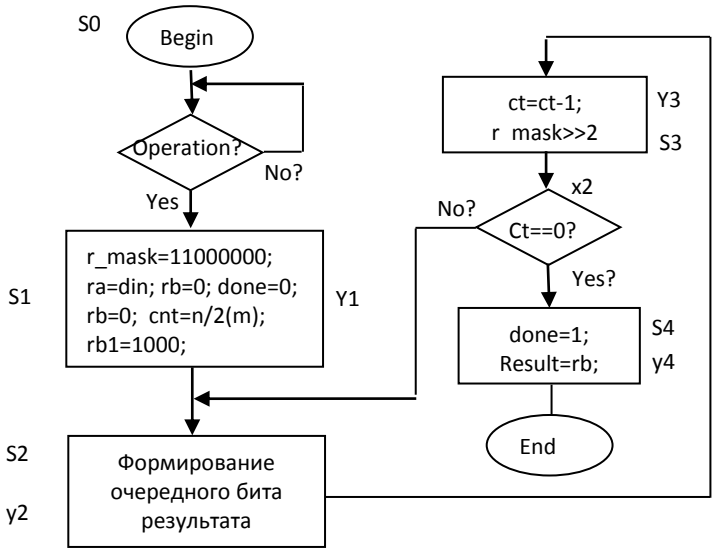


Рис. 2. Блок-схема алгоритма извлечения корня квадратного

Составлена программа на Verilog, произведен синтез с использованием системы Xilinx ISE Design Suite 14.2.

Сравнение сложности устройств, полученных в результате синтеза с использованием комплекта программ ISE Design Suite 14.2 и FPGA Spartan 3E, приведено в табл.2

Таблица 2

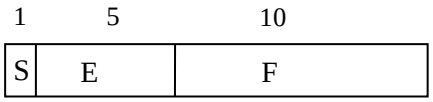
	Spartan- 3E	Комб.	С FSM	Общее число	Комб. (utilization)	С FSM (utilization)
1	Число секций (Slice)	34	20	960	3%	2%
2	Число 4-LUT	68	20	1920	3%	1%
3	Число множительных устройств	4	1	4	100%	25%
4	Число блоков I/O	24	16	66	36%	24%

Реализация устройства в виде комбинационной схемы требует значительно больше оборудования - почти в три раза. Как видно из таблицы, число секций, требуемых для комбинационной схемы, на 17 больше, чем для схемы с FSM. Число LUT больше на 38.

По быстродействию схемы не сравнимы - 1 такт против n/2 (четырех). После синтеза устройство было загружено в устройство NI Digital Electronics FPGA board.

Извлечение квадратного корня из числа с плавающей запятой.

Задано число с плавающей запятой в формате half precision (Standard IEEE-754-2008)



$A=(-1)^S\times 2^{E-15}\times 1.F$

▪ Если порядок исходного числа является четным числом, то порядок результата будет равен порядку исходного числа, деленному на 2.

▪ Если порядок – число нечетное, то его уменьшают на единицу и делят на 2. При этом мантиссу сдвигают на 1 бит влево, чтобы значение числа не изменилось.

▪ Мантисса результата определяется аналогично определению корня целого числа. Отличие в том, что мантиссу делят на группы по 2 бита, начиная с запятой.

▪ Так как корень из дроби равен корню из числителя, деленному на корень из знаменателя, то точный корень из несократимой дроби не может быть найден в том случае, если его нельзя извлечь из числителя или из знаменателя. Например, из дробей 4/5, 8/11 и 13/15 нельзя извлечь точный корень, так как в первой дроби нельзя его извлечь из знаменателя, во второй — из числителя и в третьей — ни из числителя, ни из знаменателя.

▪ Из таких чисел, из которых нельзя извлечь точный корень, можно извлекать лишь приближенные корни.

Пример: S=0; E-15=5; m=1,1001001110;

Так как порядок является нечетным числом, уменьшим его на 1. При этом порядок результата будет равен 2, а удвоенная мантисса (сдвинутая на один бит влево) - m'=11,001001110.

11, 00 10 01 11 00		$m_0=1$
1,1x1,1=10,01;	10,01<11, 00;	$m_{-1}=1$
1,11x1,11=11,0001;	11,0001<11,0010;	$m_{-2}=1$
1,111x1,111=11,100001;	11,100001>11, 001001;	$m_{-3}=0$
1,1101x1,1101=10,01001001;	10,010001001<11, 00100111;	$m_{-4}=1$
1,11011x1,11011=111000001111;	11,1000010000>11, 00100111;	$m_{-5}=0$
1,110111x1,110111=11,011001000001;	11,011001000001>11,0010011100;	$m_{-6}=1$;

Для получения оставшихся 4 цифр добавляем в исходную мантиссу справа в каждом шаге алгоритма по два нуля.

После получения m_{-10} производится округление. Для этого определяется еще одна цифра мантиссы m_{-11} и прибавляется к предыдущему результату.

Рассмотрим случай, когда порядок является отрицательным числом.

Пусть E-15= -4. m=1,001. Порядок результата E-15= -2.

Мантисса исходного числа m=01,0010;

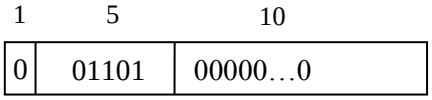
		$m_0=1$
1,1x1,1=10,01	10,01>01,00;	$m_{-1}=0$
1,01x1,01=1,1001	1,1001>01,0010	$m_{-2}=0$
1,001x1,001=1,010001	1,010001>01,0010	$m_{-3}=0$
1,0001x1,0001=1,00100001	1,00100001>01,0010	$m_{-4}=0$

. . . .

При продолжении выполнения этой процедуры получаем $m_{-10}=0$.

Результатом будет m=1,0000…0.

Результатом операции будет порядок P= -2. Смещенный порядок E=15-2=13, в двоичной системе E=011101, m=1,00000000. Ниже представлено число в формате half-precision.



$A \times 2^{E-15} \times 1.F = 1/4$

Если исходное число является точным квадратом, то приведенный алгоритм позволяет получить точное значение корня.

Структура устройства представлена на рис.3.

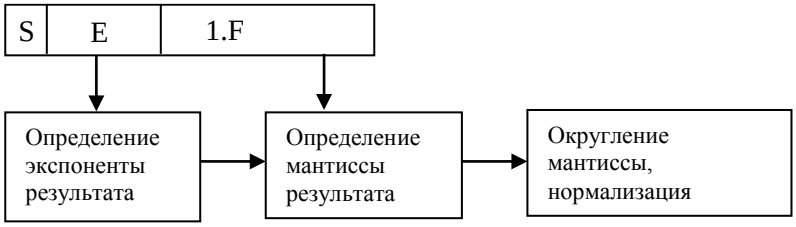


Рис. 3. Структура устройства

Операция square root широко используется в SSE и AVX расширениях системы команд современных процессоров.

Составлена программа на языке Verilog и синтезировано устройство для FPGA Spartan-3E. В результате для реализации потребовалось 86 4-входовых LUT (1%), число секций -49 (1%) , число умножителей -20 (25%).

Последовательность синтеза схемы на FPGA

1. Описание устройства (в графической форме или HDL). Симуляция.
2. Логический синтез. Получение логической схемы (gate-level netlist).
3. Имплементация проекта. Отображение логики в элементы FPGA – CLB и IOB (Input-Output blocks - интерфейсные блоки).

При этом осуществляется оптимизация проекта с учетом временных ограничений и оптимизация по занимаемой площади.

4. Получение файла Bitstream – поток битов для конфигурирования устройства (FPGA).

Результат этапа проектирования на основе FPGA – bitstream представляет собой структурное описание логической сети, заданное в формате, который является технологическим секретом производителя FPGA.

СПИСОК ЛИТЕРАТУРЫ

1. **Киселев А.** Элементы алгебры и анализа. -М., 1928 .
2. **Тарасов И.** Реконфигурируемые элементы //Открытые системы. -2011. -N5.
3. fpgach.blogspot.com/2013/10/int/html. Извлечение квадратного корня из INT.

Ա.Կ. ԹՈՒՄԱՆՅԱՆ, Ա.Գ. ՀՈՎՀԱՆՆԻՍՅԱՆ, Ա.Հ. ՀԱՄԱԶԱՍՊՅԱՆ, Ա.Գ. ՔԱՄԱԼՅԱՆ

FPGA-ի ՀԻՄԱՆ ՎՐԱ ՔԱՌԱԿՈՒՍԻ ԱՐՄԱՏԻ ԴՈՒՐՍԲԵՐՄԱՆ ՍԱՐՔԻ ՍԻՆԹԵԶՈՒՄ

Դիտարկված են ամբողջ և սահող ստորակետով թվերից քառակուսի արմատի դուրսբերման սարքի նախագծման հարցերը: Կատարված է սարքի երկու իրականացումների՝ կոմբինացիոն և հաջորդական (FSM-ի կիրառմամբ) տեսակների համեմատում, որոնք սինթեզված են XILINX ISE Design Suite 14.2. նախագծող համակարգի օգտագործմամբ:

Առանցքային բառեր. FPGA, Verilog, LUT, CLB, քառակուսի արմատ:

A.K. TUMANYAN, A.G. HOVHANNISYAN, A.H. HAMAZASPYAN, A.G. QAMALYAN

SYNTHESIS OF THE SQUARE ROOT EXTRACTION DEVICE BASED ON FPGA

Issues on designing a device for extracting the square root of an integer and floating-point numbers are considered. The two implementations of the device- the combination and the sequential type (with FSM) obtained by synthesis using a system design XILINX ISE Design Suite 14.2 are compared.

Keywords: FPGA, Verilog, LUT, CLB, square root.