

## ПРОЕКТИРОВАНИЕ МОДУЛЕЙ С ПЛАВАЮЩЕЙ ТОЧКОЙ НА ПЛИС С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА VHDL

---

**Abstract:** The structural realization of the PLD-based of a Xilinx type FPGA 32-bit module of division with a floating point appropriate to the standard IEEE-754, executed by using the behavioral description of algorithm by language VHDL is offered. The check of functioning of the module of division by a method of modeling in system ModelSim Xilinx Edition–MXE II with the help of the verifying stand executed by means of the schematic editor Engineering Capture System (ECS) and the HDL-editor, Xilinx ISE Foundation system, included in structure is realized.

**Key words:** PLD (programmable logic devices), floating point arithmetic, modeling, prototyping, verification, VHDL.

**Анотація:** Запропоновано структурну реалізацію 32-розрядного модуля ділення з плаваючою точкою, що відповідає стандарту IEEE-754, в елементному базисі ПЛИС типу FPGA фірми Xilinx, який виконано шляхом поведінкового опису алгоритму мовою VHDL. Реалізовано перевірку функціонування модуля ділення методом моделювання в системі ModelSim Xilinx Edition–MXE II за допомогою перевірного стенда, який виконано засобами схематичного редактора Engineering Capture System (ECS) та HDL-редактора, що входять до складу системи Xilinx ISE Foundation.

**Ключові слова:** ПЛИС, арифметика з плаваючою точкою, моделювання, макетування, верифікація, VHDL.

**Аннотация:** Предложена структурная реализация 32-разрядного модуля деления с плавающей точкой, соответствующего стандарту IEEE-754, в элементном базисе ПЛИС типа FPGA фирмы Xilinx, выполненная путем поведенческого описания алгоритма на языке VHDL. Реализована проверка функционирования модуля деления методом моделирования в системе ModelSim Xilinx Edition–MXE II с помощью проверочного стенда, выполненного средствами схематического редактора Engineering Capture System (ECS) и HDL-редактора, входящих в состав системы Xilinx ISE Foundation.

**Ключевые слова:** ПЛИС, арифметика с плавающей точкой, моделирование, макетирование, верификация, VHDL.

### 1. Введение

Современная элементная база на основе программируемых логических интегральных схем (ПЛИС) или Programmable Logic Devices (PLD) типа FPGA (Field Programmable Gate Array), а также новые технологии проектирования с использованием соответствующих инструментальных средств позволяют реализовать в кристалле практически любой проект цифрового устройства в требуемые сроки, имея в наличии лишь персональный компьютер и САПР ПЛИС [1]. Существенным преимуществом ПЛИС по сравнению с полузаказными и заказными большими интегральными схемами являются их универсальность и возможность быстрого программирования и перепрограммирования под заданный проект.

Разработка цифровых устройств на ПЛИС существенно упрощается за счет использования готовых технических решений, называемых IP (Intellectual Property), – блоками, ядрами (Core) [2]. IP-блоки являются модулями, параметры которых настраиваются пользователем под конкретные требования разрабатываемого проекта. В настоящее время подготовка таких технических решений многократного использования (Design reuse) сформировалась в отдельную область деятельности, осуществляемую рядом фирм для различных классов цифровых устройств. Этому в немалой степени способствовало внедрение в практику проектирования языков высокого уровня для описания аппаратуры (VHDL и Verilog) [3], а также соответствующих промышленных стандартов [4].

При решении ряда задач целесообразно использование формата представления чисел с плавающей точкой вместо формата с фиксированной точкой, обладающего такими недостатками, как ограничение диапазона представления и потеря точности при делении двух больших чисел. Функциональные IP-блоки с плавающей точкой находят широкое применение при построении

математических сопроцессоров, DSP – процессоров, встроенных арифметических сопроцессоров. В связи с этим рядом фирм разрабатываются собственные IP-блоки. Например, фирма Nallatech разработала ядра для обработки 32-разрядных операндов с плавающей точкой (согласно стандарту IEEE-754 [5]) под кристаллы серии Virtex [1]. Однако такие ядра являются коммерческим продуктом и не подлежат тиражированию.

Проблема проектирования арифметических устройств и алгоритмов для обработки операндов в формате с плавающей точкой при реализации операций сложения, умножения и деления актуальна и в настоящее время, что подтверждается докладами по данной теме на таких представительных симпозиумах, как Symposium on Field programmable gate arrays (FPGA) [6] и Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM) [7]. Стандарт IEEE-754 дает наиболее общее представление для чисел с плавающей точкой в современных компьютерах, включая Intel PC, Macintosh и большинство Unix платформ.

В данной статье предлагается подход к разработке устройства на основе современных кристаллов ПЛИС [8], осуществляющего выполнение операции деления в соответствии со стандартом IEEE-754, с использованием современных инструментальных средств для проектирования, моделирования и верификации.

В стандарте IEEE-754 операнды в формате одинарной точности представлены 32 битами – 1 бит для знака, 8 бит для порядка и 23 бита для дроби мантииссы. В связи с тем, что данный формат предполагает наличие «скрытого» старшего бита, мантиисса на самом деле содержит 24 бита.

Число с плавающей точкой представляется следующим образом:

$$X = (-1)^S \times 2^{E-127} \times 1, M ,$$

где  $S$  – знак числа;  $E$  – показатель порядка (смещенный на  $b = +127$ );  $M$  – нормализованная мантиисса ( $M \in [1,2)$ ). На рис. 1 приведен формат операнда и диапазон допустимых значений чисел.

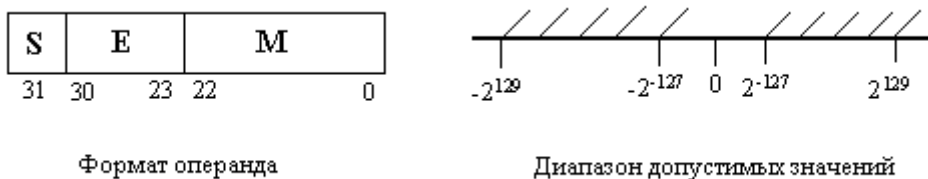


Рис. 1. Формат представления операнда и диапазон допустимых значений

Стандарт резервирует ряд специальных значений:

- машинный ноль                   “x\_00000000\_000000000000000000000000”;
- ненормализованное число “x\_00000000\_xxxxxxxxxxxxxxxxxxxxxxxx1”;
- + бесконечность               “0\_11111111\_000000000000000000000000”;
- - бесконечность               “1\_11111111\_000000000000000000000000”;
- QNaN                             “x\_11111111\_1xxxxxxxxxxxxxxxxxxxxxxxx”;
- SNaN                             “x\_11111111\_0xxxxxxxxxxxxxxxxxxxxxxxx1”.

Здесь  $x$  может принимать значение нуля или единицы. NaN (Not a Number – не число) используется для представления значения, которое не представляется вещественным числом. NaN содержит две категории: QNaN (Quiet NaN) и SNaN (Signaling NaN). QNaN – семантически обозначает неопределенные операции, SNaN – недопустимые операции.

В данной работе рассматривается подход к реализации модулей деления с плавающей точкой, которые представляют собой законченные типовые технические решения и могут быть использованы в других проектах в качестве soft cores.

Работа выполнена в рамках ДНТП "Нові вітчизняні інтелектуальні комп'ютерні засоби" Министерства образования и науки Украины.

## 2. Методология проектирования

Рассмотрим процедуру проектирования устройства, выполняющего операцию деления с плавающей точкой над 32-разрядными операндами в соответствии со стандартом IEEE-754. Деление с плавающей точкой в общем случае включает сравнение мантисс делимого и делителя, сдвиг вправо мантиссы делимого и вычисление частного. Результирующий порядок (как разность порядков делимого и делителя) вычисляется параллельно. Процедура выполнения операции деления представляется в следующем виде:

$$C = \frac{A}{B} = \frac{(-1)^{S_1} \times 2^{E_1-127} \times 1, M_1}{(-1)^{S_2} \times 2^{E_2-127} \times 1, M_2} = (-1)^{S_3} \times 2^{E_3-127} \times 1, M_3,$$

где  $E_1, E_2, E_3$  – порядки, а  $M_1, M_2, M_3$  – мантиссы соответственно первого, второго операндов и результата.

Таким образом, с учетом требований стандарта IEEE-754, проектируемый модуль деления содержит:

три входа:  $A(31:0)$ ,  $B(31:0)$  – входные операнды, **Load** – сигнал загрузки операндов (состояние '1' соответствует операции загрузки, '0' – операции деления), **clk** – последовательность синхроимпульсов;

три выхода:  $C(31:0)$  – результат деления чисел  $A$  и  $B$ , **Rdy** – сигнал готовности или окончания операции (состояние '1' соответствует окончанию операции, '0' – началу операции деления), **Error(6:0)** – регистр ошибок. Регистр может принимать следующие значения: "0000000" – отсутствие ошибок, "0000001" – операнды  $A$  и/или  $B$  имеют специальные значения, "0000011" – результат «ненормализованное число», "0000101" – результат «машинный нуль», "0001001" – результат "+ бесконечность", "0010001" – результат "- бесконечность", "0100001" – результат QNaN, "1000001" – результат SNaN.

Алгоритм операции деления с плавающей точкой (при его поведенческом описании на языке VHDL) представляется в виде конечного автомата (State machine) с 6-ю состояниями ( $R1 - R6$ ) и условиями перехода  $R1 \rightarrow R2$ ,  $R2 \rightarrow R3$  и  $R4 \rightarrow R5$  (рис. 2).

В состоянии  $R1$ , инициализированном сигналом загрузки данных  $Load$ , производится установка выхода  $Rdy$  в состояние логического нуля и анализ входных операндов  $A$  и  $B$ . Если

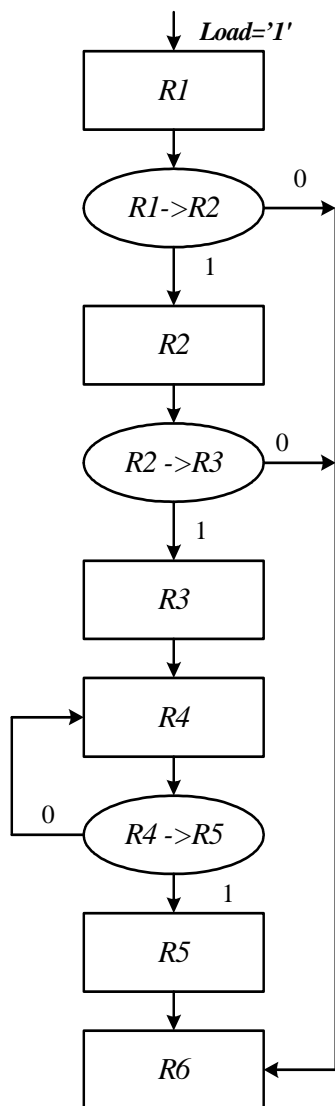


Рисунок 2

при анализе обнаружена ошибка, то в регистр ошибок записывается код "0000001" и осуществляется переход в состояние  $R6$ , в котором производится установка  $Rdy$  в состояние логической единицы и происходит окончание операции. При отсутствии ошибки в регистр ошибок записывается код "0000000" и автомат переходит в состояние  $R2$ .

В состоянии  $R2$  формируются знак и значения порядка результата деления (частного):

$$S_3 = S_1 \oplus S_2, E_3 = E_1 - E_2 + 127.$$

Если  $M_1 < M_2$ , то значение  $E_3$  уменьшается на «единицу» с установкой в '1' соответствующей внутренней переменной "flag". Далее производится проверка условия  $0 > E_3 > 255$ , при выполнении которого автомат переходит в состояние  $R6$ , а при невыполнении – в состояние  $R3$ .

В состояниях  $R3$  и  $R4$ , связанных безусловным переходом, осуществляется формирование мантиссы результата при делении  $M_1/M_2$ .

В состоянии  $R3$ : если переменная "flag" = '0', то производится операция  $(M_1 - M_2)$ , если "flag" = '1' ( $M_1 < M_2$ ), то  $(2M_1 - M_2)$ . Значение полученной разности удваивается и присваивается переменной  $M_1$ . Так как мантисса содержит 23 разряда, то переменной, характеризующей количество циклов вычитания, присваивается значение  $(i = 23)$  и осуществляется переход в состояние  $R4$ .

В состоянии  $R4$  в каждом цикле вычитания, если номер цикла больше или равен '0', производится сравнение  $M_1$  с  $M_2$ . Если  $M_1$  больше или равно  $M_2$ , то разряду  $M_3$ , определяемому в данном цикле, присваивается значение '1' и выполняется операция  $(M_1 - M_2)$ . В противном случае вычитание не производится и соответствующему разряду  $M_3$  присваивается нулевое значение. При выполнении каждого цикла предыдущее значение переменной  $M_1$  удваивается, а номер цикла уменьшается на единицу. После того, как требуемое количество циклов будет исчерпано  $(i < 0)$ , произойдет переход автомата в состояние  $R5$ .

В состоянии *R5* производится операция симметричного округления мантиссы, ее нормализация и запись знака, порядка и мантиссы результата деления в выходной порт (если полученный результат не является специальным значением). В противном случае осуществляется запись соответствующего кода в регистр ошибок. Далее, независимо от проведенного анализа, автомат переходит в состояние *R6*.

Описание модуля деления, выполненное на языке VHDL, синтезировано и реализовано в кристалле серии Spartan II - 2S50PQ208-6 с помощью средства XST VHDL (Xilinx Synthesis Technology) [9] системы проектирования Xilinx ISE (Integrated Synthesis Environment) Foundation). В таблице приведены оценки аппаратных и временных параметров разработанного модуля.

Таблица. 1. Оценки временных и аппаратных параметров

Оценка аппаратных ресурсов после выполнения процедуры синтеза	Слайсы (Slices)	Триггеры (Slice Flip Flops)	4-входовые LUT	Входы/выходы IOB
	302 (39%)	174 (11%)	553 (36%)	105 (75%)
Оценка аппаратных ресурсов и быстродействия после выполнения процедуры реализации (MAP и Place&Route)	Слайсы (Slices)		Синхросигналы Clk Период (нс) / Частота (МГц)	
	323 (42%)		9,91/100	

**Примечание.** В скобках после указания количества слайсов, триггеров, LUT и IOB указан объем затраченных ресурсов в процентах к общему объему данного типа кристалла.

### 3. Верификация модуля деления

Проверка функционирования модуля деления осуществлена методом моделирования в системе ModelSim Xilinx Edition–MXE II [10]. Наиболее трудоемким процессом при выполнении моделирования является создание файла входных воздействий, т.е. генерация сигналов, поступающих на входы проверяемого модуля. HDL-технология позволяет генерировать проверяющие тесты, называемые Test Bench, на основе VHDL-языка. Для данного модуля задача осложняется тем, что система моделирования работает с целыми числами, представленными в формате с фиксированной точкой. Поскольку преобразование чисел из одного формата в другой является трудоемкой операцией, целесообразно это преобразование проводить с помощью соответствующих модулей, образующих стенд для проверки модуля деления. На рис. 3 представлена схема такого стенда, выполненная средствами схематического редактора Engineering Capture System (ECS), входящего в состав системы Xilinx ISE Foundation.

На вход стенда поступают два целых 24-разрядных числа (*din\_a* – делимое, *din\_b* – делитель), которые преобразуются модулями *int\_fp* из формата представления с фиксированной точкой в формат с плавающей точкой, а результат деления этих чисел на выходе модуля деления *div\_32\_format* преобразуется модулем *fp\_int* из формата с плавающей точкой в формат с фиксированной точкой. В качестве делимого задается константа, равная единице, в качестве делителя – последовательность чисел 1, 2, 3, ..., 16. На рис.4 приведены временные диаграммы (Waveform) деления числа 1000000 на последовательность чисел 1, 2, 3, ..., 16 (верхняя часть рисунка) и последовательности чисел  $1^2, 2^2, 3^2, \dots, 16^2$  на 1, 2, 3, ..., 16 (нижняя часть рисунка).

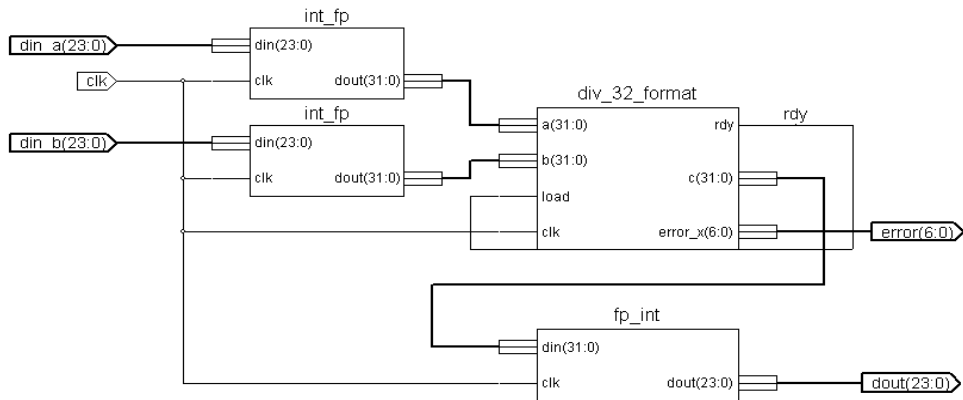


Рис. 3. Структура стенда для проверки модуля деления

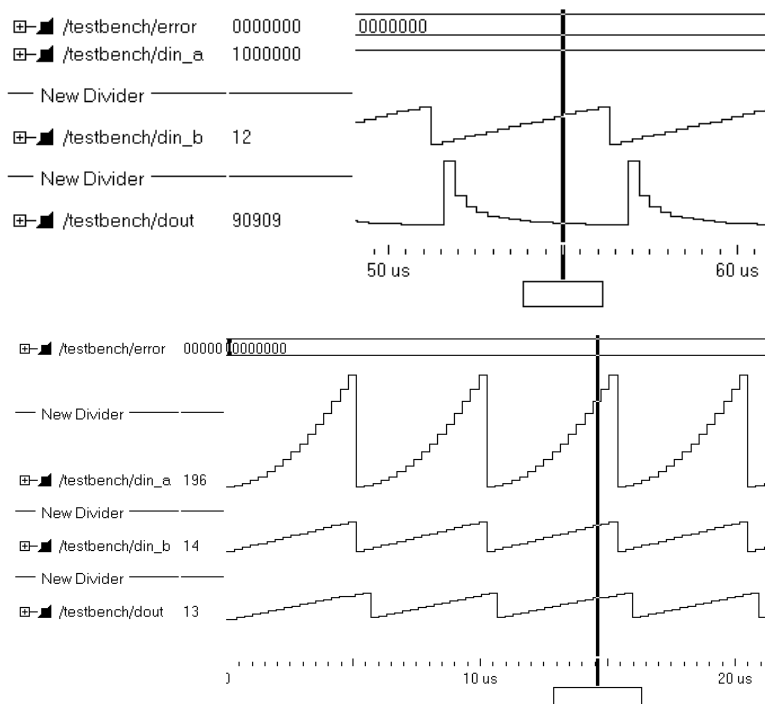


Рис. 4. Временные диаграммы

Для временной диаграммы, представленной в верхней части рисунка, входные сигналы описываются в виде трех процессов: для *clk*, числа *din\_a* и числа *din\_b*.

Аналогично описываются входные воздействия для временной диаграммы, представленной в нижней части рисунка. При составлении реальных тестов массив входных воздействий не ограничивается выборкой из 16 значений и обязательно включает в себя числа на границах диапазона представления. Также для генерации входных воздействий может быть использован генератор случайных чисел. После чего производится сравнение двух массивов чисел: результатов, полученных при моделировании, и результатов, полученных при работе программной модели.

```

Tb1 : process
begin
  clk <= '0'; wait for 5 ns;
  clk <= '1'; wait for 5 ns;

```

```

end process;
Tb2 : process
begin din_a<="000011110100001001000000";
wait;
end process;
Tb3 : process
variable temp1: integer range 0 to 33554431;
begin
temp1:=1;
for l in 1 to 16 loop
din_b<=conv_std_logic_vector(temp1,24); wait
for 300 ns;
temp1 := l + 1;
end loop;
end process;

```

#### 4. Выводы

Использование инструментальных средств FPGA Express и программ размещения и трассировки системы Xilinx Foundation Series позволило осуществить синтез и реализацию модуля деления с плавающей точкой в кристалле XCV50CS144-6, который занимает 42% аппаратных ресурсов данного кристалла (323 слайса) и может функционировать с тактовой частотой 100 МГц.

Синтезированный модуль деления верифицирован методом моделирования в системе ModelSim Xilinx Edition–MXE II с помощью стенда для проверки, что подтверждает правильность функционирования данного модуля.

Полученные результаты будут использованы при разработке сложных вычислительных устройств с использованием типовых технических решений (в виде библиотечных элементов IP Core) функциональных блоков с плавающей точкой в качестве составных компонентов для реализации таких устройств на одном кристалле. Такие типовые решения в совокупности представляют собой «открытую» библиотеку файлов конфигураций, входящих в состав реконфигурируемой вычислительной системы с виртуальной архитектурой.

#### СПИСОК ЛИТЕРАТУРЫ

1. Xilinx Data Book 2004 / Available at <http://support.xilinx.com/partinfo/databook.htm>.
2. Палагин А.В., Опанасенко В.Н., Сахарин В.Г. Системы верификации на основе реконфигурируемых устройств // Математичні машини і системи. – 2004. – № 2. – С. 100 – 113.
3. VHDL'93. IEEE Standard VHDL Language Reference Manual. IEEE Std 1076–1993.
4. Hollash S. IEEE Standard 754 Floating Point Numbers / S. Hollash/ Available at [http://IEEE/IEEE Standard 754 Floating-Point.htm](http://IEEE/IEEE%20Standard%20754%20Floating-Point.htm).
5. IEEE 754 compatible floating point cores for Virtex®-II FPGAs / Available at <http://www.nallatech.com>.
6. Doss C., Riley R.L. FPGA-based implementation of single-precision exponential unit // Proc. of the 12th International symposium on Field programmable gate arrays (FPGA). – Monterey, California. – 2004. – 22–24 February. – P. 248.
7. Krueger S.D., Seidel P.-M. Design of an On-Line IEEE Floating-Point Addition Unit for FPGAs // Proc. of 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM '04) Marriott at Napa Valley, Napa, California. – 2004. – 20–23 April. – P. 229 – 238.
8. Палагин А.В., Опанасенко В.Н., Сахарин В.Г. Проектирование цифровых устройств на кристаллах ПЛИС с использованием Core – ядер // Вестник СевГТУ: Сб. научных трудов. – Севастополь: 2004. – Вып. 57. – С. 33 – 47.
9. Xilinx Synthesis Technology (XST). User Guide / Available at <http://www.xilinx.com>.
10. Synthesis and Simulation. Design Guide / Available at <http://www.xilinx.com>.