УДК 004.94

Моделирование внутренних операций процессорных элементов

Р.В. Мальчева, Т.В. Завадская
Донецкий национальный технический университет
raisa_malcheva@donntu.org

Мальчева Р.В., Завадская Т.В. Моделирование внутренних операций процессорных элементов. *Выполнен анализ необходимости разработки демонстрационных, обучающих и имитационных моделей внутри процессорных операций. Приведены примеры моделей различного уровня. Выполнено проектирование устройств с использованием языков описания аппаратуры HDL и Verilog.*

Ключевые слова: *демонстрационная модель, средства обучения, проектирование, HDL, Verilog, исследования.*

Введение

Стремительное развитие компьютерных и, в особенности, микропроцессорных технологий привело к отсутствию доступа к большинству компонентов вычислительных устройств (регистров, сумматоров, вентильных схем и т. п.), расположенных внутри одной или нескольких больших интегральных схем (БИС), что, соответственно, усложнило изучение внутренних процессов и операций.

В связи с этим все более актуальной становится задача разработки специальных моделей: физических или функциональных [1], позволяющих как демонстрировать внутреннее представление данных и выполнение операций, так и поддерживать различные стадии процесса проектирования процессорных элементов.

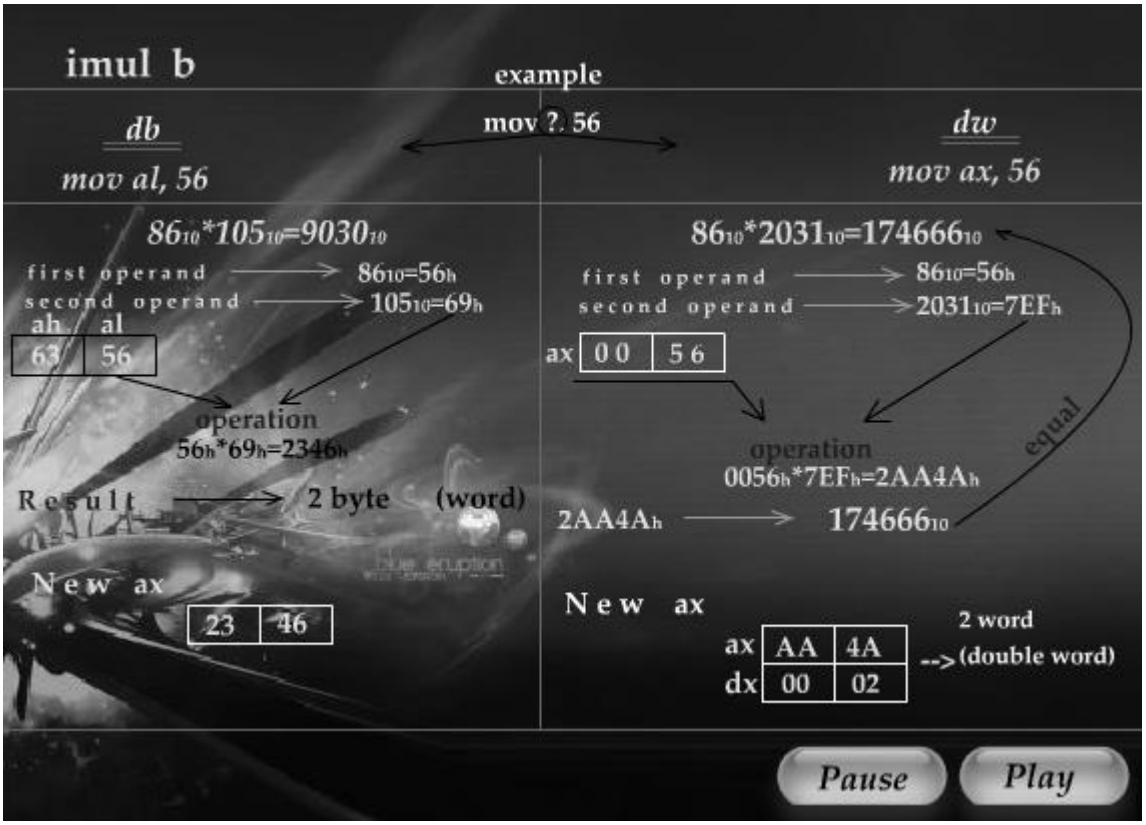
Подходы к обеспечению процесса обучения

Обеспечение образовательного процесса по дисциплинам, связанным с изучением архитектуры компьютеров, предусматривает создание учебных лабораторных комплексов на базе микропроцессорных комплектов или создание компьютерных моделей процессорных и других цифровых устройств [1]. Первый путь позволял проводить физическое исследование цифровых устройств, но требовал периодической замены лабораторной базы. Так за прошедшие годы на кафедре были введены в строй учебные лабораторные комплексы на микропроцессорных комплектах К580, К1804 и др., на которые истрачено достаточно много средств. Второй путь предполагал создание программной модели исследуемого процессора, цифрового устройства или вычислительной машины в целом, реализуемой в некоторой инструментальной среде. Такой моделью стала персональная вычислительная машины, на которой легко изменять как структуру, так и программу изучения функционирования процессора и любого цифрового устройства [1].

На кафедре компьютерной инженерии ДонНТУ уже более 30 лет применяется и второй подход - проектирование гипотетических вычислительных машин для изучения с их помощью архитектуры компьютеров, а также использования самого процесса их разработки в качестве средства обучения.

Демонстрация процессорных операций

В последние десятилетия с появлением мультимедийных технологий изменились и презентационные свойства моделей [2]. Современные средства существенно облегчают процесс обучения через реализацию одного из фундаментальных его методов – наглядности. Наиболее удобной для создания демонстраций является технология Flash, которая поддерживает не только все презентационные возможности, но и позволяет демонстрировать работу некоторого процесса шаг за шагом с заданной частотой смены кадра [3]. Как показывает практика подготовки специалистов направления компьютерной инженерии, особое затруднение у студентов вызывает понимание машинного (внутреннего) представления данных и их обработки. Для повышения эффективности преподавания дисциплин, связанных с изучением основ информатики и особенностей выполнения машинных команд, был подготовлен значительный объем демонстрационных роликов [4]. В качестве примера на рис. 1 приведены слайды демонстрационного ролика выбора внутренних регистров процессора и формирования результата при выполнении операции целочисленного умножения для 8-мибитных и 16-тиразрядных операндов.



а) б)

Рисунок 1 - Демонстрация выбора внутренних регистров процессора и формирования результата при выполнении операции целочисленного умножения для 8-мибитных (а) и 16-тиразрядных (б) операндов

Анимационные ролики используются не только во время чтения лекций. Разработаны Web-версии учебных дисциплин, со страниц которых студенты могут просматривать демонстрационные ролики, а также проходить самотестирование изученного материала.

Разработка моделей функционирования устройств

Как и во многих университетах [1, 5], в качестве заданий по курсовому проектированию применяется разработка студентами моделей (рис. 2) гипотетических компьютеров - упрощенных архитектур наиболее популярных промышленных процессоров (рис. 3).

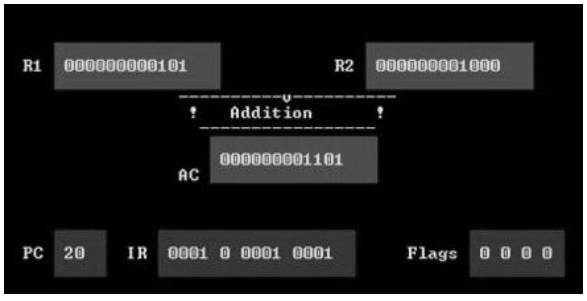


Рисунок 2 – Простейшая программная модель выполнения команды «сложение» в одноадресной гипотетической машине

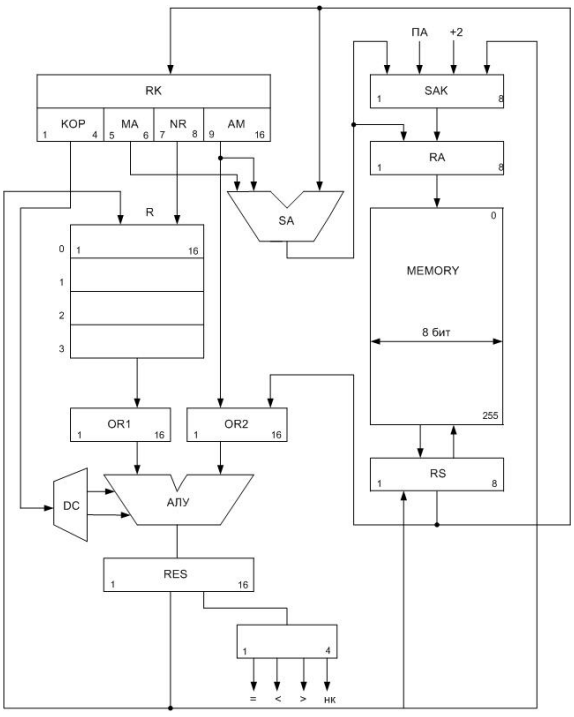


Рисунок 3 – Вариант архитектуры одноадресной гипотетической машины

Для ознакомления студентов с принципом работы одно- двух- и трехадресной вычислительной машины созданы специальные

Flash-фильмы [6]. Разработка программных моделей применяется и для изучения функционирования различных аналого-цифровых преобразователей. На рис. 4 приведен пример моделирования логики работы аналого-цифрового преобразователя поразрядного уравновешивания.

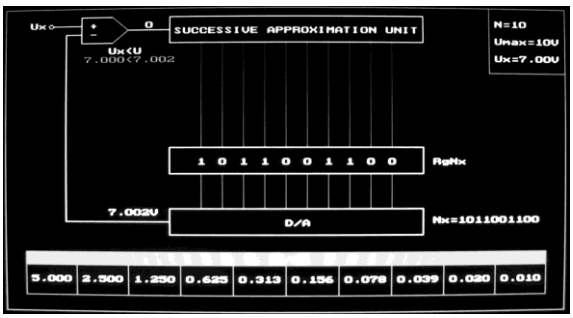


Рисунок 4 – Модель АЦП поразрядного уравновешивания

Примеры проектирования устройств

Проектирование устройств выполняется с использованием таких языков описания аппаратуры, как HDL, Verilog и др. [7-10].

Рассмотрим пример разработки процессорного элемента для вычисления скалярного произведения *r* двух векторов в соответствии с математическим выражением (1):

$$r = x_1 \cdot x_2 + y_1 \cdot y_2 + z_1 \cdot z_2 \ , \qquad (1)$$

где $\{ x_1, y_1, z_1 \}$ - компоненты 1-го вектора;

$\{ x_2, y_2, z_2 \}$ - компоненты 2-го вектора.

Приведем код для вычисления скалярного произведения на языке Verilog:

```
module top( //выходы элемента
 output wire[15:0] r,
 output wire vis,  // флаг видимости
 // входные данные
    input wire[15:0] x1,
    input wire[15:0] y1,
    input wire[15:0] z1,
        input wire[15:0] x2,
    input wire[15:0] y2,
    input wire[15:0] z2 );
// вычисление скалярного произведения
    assign r = x1*x2 + y1*y2 + z1*z2;
Endmodule
```

Вычисление формулы в программе автоматически разбивается средой на отдельные действия (x1*x2, y1*y2, …), которые выполняются параллельно. Схема полученного устройства изображена на рис. 5. Чтобы показать параллельность вычислений, в код модели были добавлены промежуточные переменные *m1, m2, m3, s1* и внесены временные задержки следующим образом:

```
wire[15:0] m1;
wire[15:0] m2;
wire[15:0] m3;
wire[15:0] s1;
assign #20 m1 = x1*x2;
assign #20 m2 = y1*y2;
assign #20 m3 = z1*z2;
assign #5 s1 = m1+m2;
assign #5 r = s1+m3;
```

При моделировании получены следующие временные диаграммы (рис. 6).

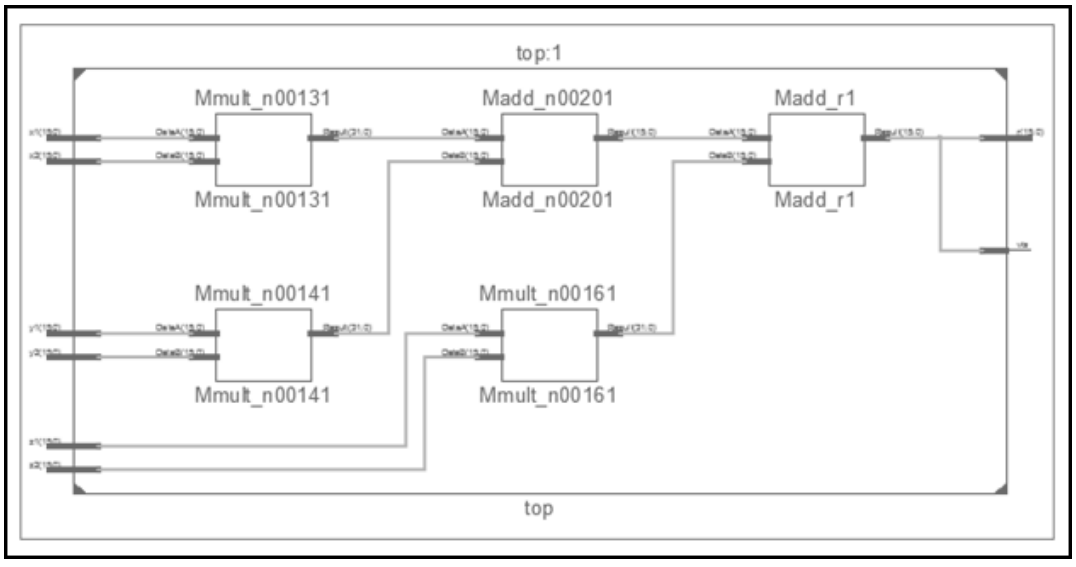


Рисунок 5 – Функциональная схема устройства для вычисления скалярного произведения

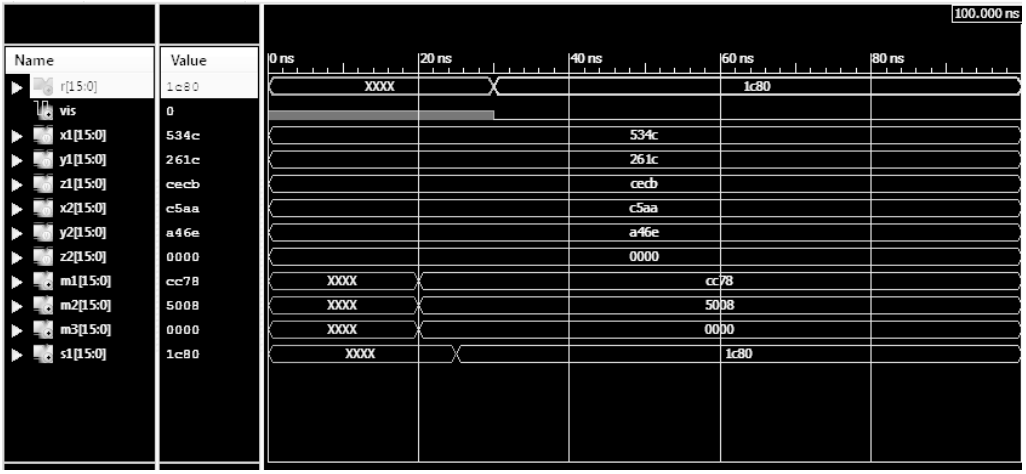


Рисунок 6 – Результаты моделирования устройства для вычисления скалярного произведения

Для описания более детального проектирования рассмотрим пример реализации в виде VHDL модели устройства умножения. Оно (рис. 7) состоит из умножителя, элемента И-2-ИЛИ, D-тригера, DC-тригера. При нажатии кнопки Reset, логический «0» поступает на DC триггер и на блок умножения, происходит обнуления регистра AC, записывается множимое и множитель, происходит умножение. D-триггер срабатывает по сигналу ready умножителя, он необходим для буферизации данных.

Умножитель (рис. 8) включает:

MC - 8 разрядный регистр множимого. Загрузка множимого осуществляется по команде Cload_mc с управляющего автомата.

MR - 8 разрядный регистр множителя. По команде Cload_mr в регистр загружается множитель, по команде Cshift_right_mr происходит логический сдвиг вправо. Нулевой разряд является признаком коррекции произведения.

AC - 16 разрядный регистр результата. Имеет логический сдвигатель вправо (Cshift_right_ac), команду сброса (Creset), команду загрузки результата с АЛУ (Calu).

ALU - 8 разрядное арифметико-логическое устройство, содержит сумматор (Cadd/sub := 1) и вычитатель (Cadd/sub := 0).

CT - счетчик, предназначенный для подсчета операций. Имеет признак переполнения CTF. Команда Creset_ct сбрасывает счетчик в исходное состояние, а Cinc - инкрементирует содержимое счетчика.

Control Unit - управляющий автомат.

VHDL модель MC:

```
process (CLK)
begin
//если восходящий вход CLK
  if (rising_edge(CLK)) then
    //и если сигнал Enable
    if (EN='1') then
      //то записать данные
        Q <= D;
    end if;
  end if;
end process;
```

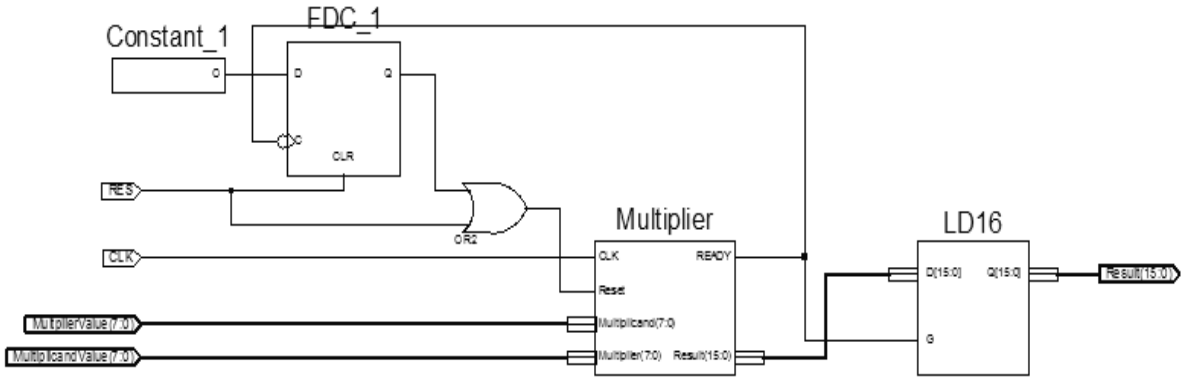


Рисунок 7 – VHDL модель устройства умножения

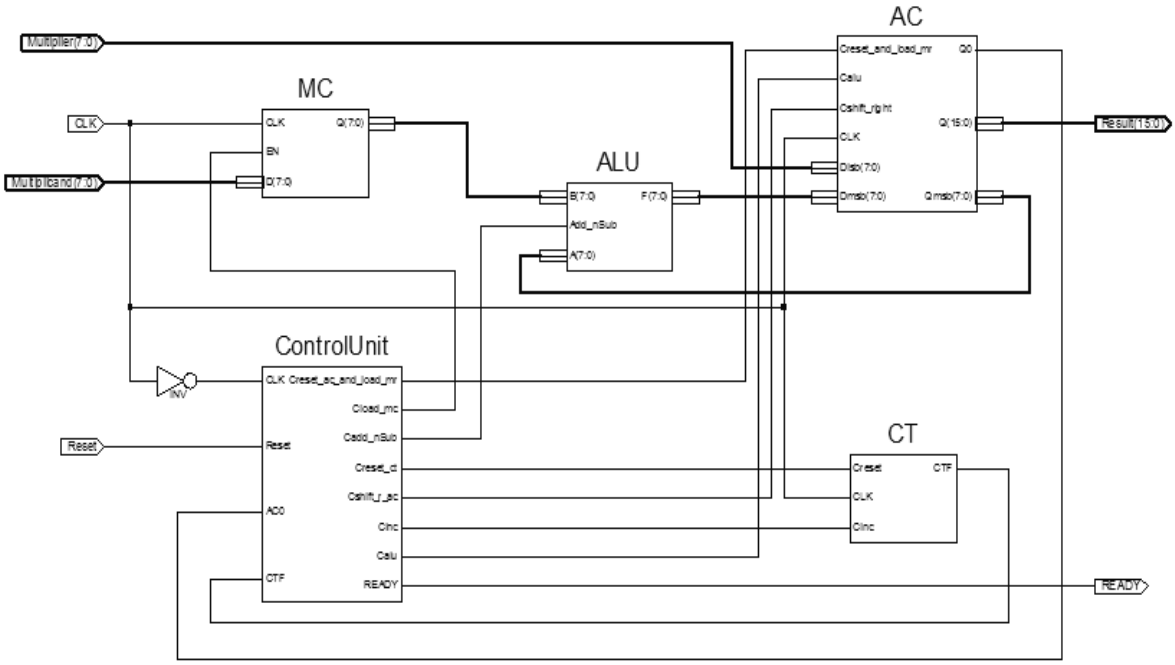


Рисунок 8 – VHDL модель умножителя

Блок АЛУ включает в себя:

- инкрементор:

```
process (A)
  begin
    I <= A+1;
end process;
```

- инвертор;
- сумматор:

```
process (A,B)
variable S:STD_LOGIC_VECTOR (7 downto 0);
  begin
    S := A + B;
    ModS <= S(6 downto 0);
    SignS  <= S(7);
end process;
```

Для формирования произведения используется обычный сумматор, а после инвертируется знаковый разряд. Для выполнения операции вычитания необходимо поменять знак у второго операнда (B) и провести сложение. Смена знака производится путем инверсии операнда, а затем инкрементации полученного значения. Счетчик, реализованный в виде VHDL модели, включает в себя 4-х разрядный счетчик и элемент 3И и реализует следующую сущность:

```
entity CB4CE_HXILINX_CT is
port (
    Q0 : out STD_LOGIC;
    Q1 : out STD_LOGIC;
    Q2 : out STD_LOGIC;
    Q3 : out STD_LOGIC;
    C  : in STD_LOGIC;
    CE : in STD_LOGIC;
    CLR: in STD_LOGIC
    );
end CB4CE_HXILINX_CT;
```

Реализация счетчика выглядит следующим образом:

```
architecture Behavioral of CB4CE_HXILINX_CT
is
 signal COUNT : STD_LOGIC_VECTOR(3 downto 0)
:= (others => '0');
    constant TERMINAL_COUNT :
STD_LOGIC_VECTOR(3 downto 0) := (others =>
'1');
    begin
    process(C, CLR)
    begin
      if (CLR='1') then
        COUNT <= (others => '0');
      elsif (C'event and C = '1') then
        if (CE='1') then
          COUNT <= COUNT+1;
        end if;
      end if;
    end process;
      Q3 <= COUNT(3); Q2 <= COUNT(2);
      Q1 <= COUNT(1); Q0 <= COUNT(0);
 end Behavioral;
```

По сигналу CRL значение счетчика сбрасывается в исходное состояние. Наращивание счетчика происходит по фронту сигнала CE. Для данной задачи необходим 3-х разрядный счетчик, поэтому сигнал переполнения добавляется логическим элементом 3И.

Регистр АС реализован в виде следующего кода:

```
process (CLK,RESULT)
begin
  if (rising_edge(CLK)) then
    if (Creset_and_load_mr ='1') then
      RESULT(15 downto 8) <= "10000000";
     //Сброс старшей части
      RESULT(7 downto 0) <= Dlsb;
     //Записываем множимое в младшую часть
     end if;
```

```
        if (Calu = '1') then
          RESULT(15 downto 8) <= Dmsb;
        end if;
        if (Cshift_right = '1') then
        //логический сдвиг
          RESULT(15) <= RESULT (15);
          RESULT(14) <= not (RESULT(15));
          RESULT(13 downto 0) <= RESULT(14
downto 1);
        end if;
      end if;
      Qmsb <= RESULT (15 downto 8);
        //Множимое
      Q0 <= RESULT (0);
        //Младший бит
      Q <= RESULT;
// Запись результата
   end process;
```

Блок Control Unit представлен в виде VHDL-модели и реализует алгоритм операции умножения в положительном нуле. В общем виде модель выглядит следующим образом:

```
    process (CLK,Reset)
     begin
      if (Reset = '1') then //Сигнал сброса
          state <= RST;
        //Если фронт CLK
          elsif rising edge(CLK) then
           case state is
           //Проверка текущего состояния
              when RST => … //состояние
сброса
           //проверка нулевого бита множителя
              when BIT CHECK => …
           //сдвиг регистра AC
             when SHIFT => …
           //проверка переполнения и проверка
         //знакового бита, после переполнения
              when CTF AND SIGNBIT CHECK => …
         //сдвиг, если знаковый бит равен «1»
              when LAST SHIFT => …
       //состояние готовности
          when RDY => …
         end case;
      end if;
   end process;
```

Плата устройства умножения разработана с использованием комплексной системы проектирования высокоскоростных устройств altium designer 16 [11]. Трассировка платы приведена на рис. 9. Для наглядности спроектирована трехмерная модель платы, показанная на рис. 10.

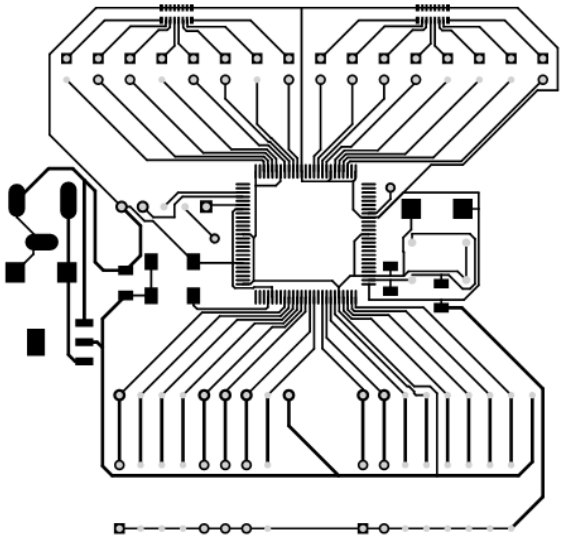
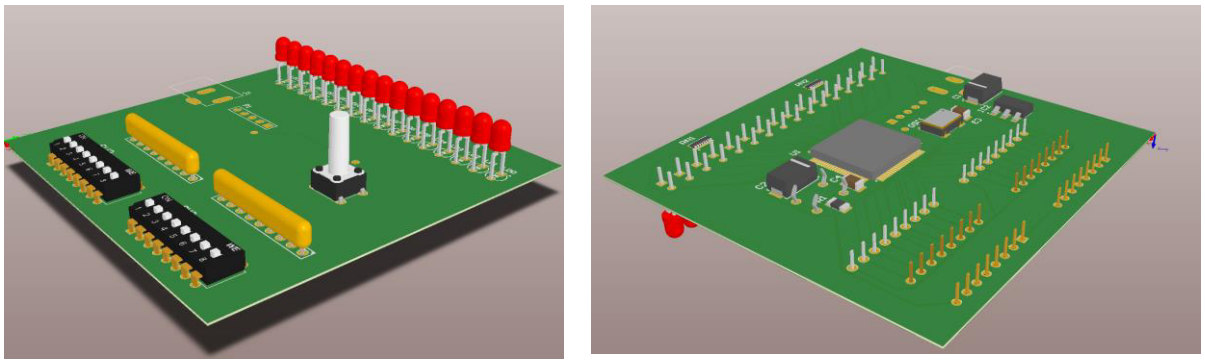


Рисунок 9 – Трассировка платы устройства умножения



а) б)

Рисунок 9 – Трехмерная модель платы устройства умножения: а) вид сверху; б) нижняя панель платы

Т.о. кратко прослежены основные стадии процесса проектирования устройства.

Заключение

Для обеспечения дисциплин, связанных с изучением архитектуры и составных элементов компьютеров, необходимо применять широкий спектр демонстрационных и обучающих средств. Для их создания используются различные современные технологии, такие как Power Point и Flash.

Для изучения основ проектирования элементов вычислительных систем и выполнения соответствующих курсовых и дипломных проектов необходимо использовать языки описания аппаратуры, такие как HDL, Verilog и др. Модели, созданные на языках проектирования VHDL или Verilog, можно не только запускать в средах проектирования, но и

загрузить в FPGA плату и подключить к нему устройства ввода и вывода. Существует множество отладочных FPGA стендов, которые уже имеют встроенные устройства ввода, такие, как кнопки и переключатели; и вывода – световые индикаторы, экран. Также в таких стендах имеются различные порты ввода/вывода, к которым можно подключить периферийные устройства.

Особое место в процессе организации учебного процесса занимает создание простых гипотетических машин, обладающих типичными чертами многих конкретных процессоров. Знание принципов построения и функционирования таких машин является хорошей базой для дальнейшего изучения и применения вычислительных систем.

Список использованной литературы

1. Кириллов В.В. Архитектура базовой ЭВМ. – СПб: СПбГУ ИТМО, 2010. – 144 с.

2. Malcheva R. Applying Internet technologies to improve the perception of lectures // Proceedings of 3d Congress EE. - Glasgow, 2002. - PP. 348-349.

3. Macromedia Flash MX 2004 ActionScript 2.0. Справочник разработчика: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. - 896 с.

4. Мальчева Р.В. Использование видеороликов и игр при обучении иностранных студентов направления "Компьютерная инженерия" // Збірка праць V науково-методичної конференції “Проблеми і шляхи вдосконалення науково-методичної та навчально-виховної роботи в ДонНТУ”. – Донецьк: ДонНТУ, 2013.

5. Baranov S. Digital System Design. In book “Design of Digital Systems and Devices”. Series: Lecture Notes in Electrical Engineering. – Springer-Verlag Berlin Heildelberg, Vol. 79, 2011. – P. 3- 41.

6. Мальчева Р.В., Сбитнева М. С. Интерактивная анимация как метод повышения эффективности изучения архитектуры компьютеров // Інформатика та комп’ютерні технології. Матеріали VI міжнародної науково-технічної конференції студентів, аспірантів та молодих науковців. - Донецьк, ДонНТУ, 2010.- Т.2. - С. 57–61.

7. Rawski M., Tomaszewicz P., Borowik G. Logic Synthesis Method of Digital Circuits Designed for Implementation with Embedded Memory Blocks of FPGAs. In book “Design of Digital Systems and Devices”. Series: Lecture Notes in Electrical Engineering. - Springer-Verlag Berlin Heildelberg, Vol.79, 2011.– P. 121-144.

8. Хаханов В.И., Хаханова И.В., Литвинова Е.И., Гузь О.А. Проектирование и верификация цифровых систем на кристаллах. Verilog & System Verilog. – Харьков: ХНУРЭ, 2010. – 528 с.

9. Malcheva R., Naaem H. Development of the Data Transferring System Using SoC // European Scientific Journal, 2014. - Vol. 10. - N 7. - PP. 168-172.

10. Grout I. Digital Systems Design with FPGAs and CPLDs. – Amsterdam: Elsevier, 2008. – 406 pp.

11. http://dugtor.ru/programmy/raznoe/25293-altium-designer-16111-build-255-multi-ru.html

Поступила в редколлегию 12.2016

Malcheva R.V., Zavadskaya T.V. Simulation of the internal operations of the processing elements. *The need for demonstration, training and simulation of the internal operations of the processing elements is analyzed. Examples of models at different functional levels are demonstrated. The processing devices are implemented using hardware description languages HDL and Verilog.*

Keywords*: model for demonstration, learning tools, designing, HDL, Verilog, investigation.*

Мальчева Р.В., Завадська Т.В. Моделювання внутрішніх операцій процесорних елементів. *Виконаний аналіз необхідності розробці моделей, що демонструють, обучають та імітують виконання операцій процесорів. Наведені приклади моделей різного рівня. Виконано проектування пристроїв з використанням мов опису апаратури HDL и Verilog.*

Ключові слова: *демонстраційна модель, засоби навчання, проектування, HDL, Verilog, дослідження.*

Статья поступила в редакцию 20.11.2016
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом