

ИНТЕРФЕЙС ДИСПЛЕЯ ДЛЯ УЧЕБНОГО СТЕНДА НА ОСНОВЕ ПРОГРАММИРУЕМОЙ ЛОГИЧЕСКОЙ ИНТЕГРАЛЬНОЙ СХЕМЫ

Болдырев А.В.

*Федеральное государственное бюджетное образовательное учреждение высшего образования
"Донской государственный технический университет Ростов-на-Дону", Ростов-на-Дону,
Российская Федерация, e-mail: avb49@aaanet.ru*

Статья посвящена актуальной задаче обучения студентов технических специальностей вопросам программирования электронных устройств на основе языка описания аппаратуры Verilog. Цель исследования – разработка и верификация образовательного проекта на примере интерфейса дисплея в составе учебного стенда. В рамках исследования проведен анализ архитектуры учебного стенда и протокола обмена данными между программируемой логической интегральной схемой и дисплеем. Особое внимание удалено обоснованию применения 4-разрядного режима передачи данных, позволяющего оптимизировать использование выводов микросхемы. Проект реализован на языке Verilog, моделирование и отладка произведены в среде Quartus II, для экспериментальной проверки использовалась программируемая логическая интегральная схема стендла. Разработанный модуль драйвера выполняет инициализацию дисплея, преобразует 8-битные данные в последовательность полубайтов для передачи по 4-разрядной шине и управляет основными сигналами дисплея. Для тестирования создан модуль, генерирующий 8-битные коды символов. Экспериментальная проверка модуля драйвера подтвердила корректность работы интерфейса. Полученное решение упрощает отладку и моделирование схем на учебном стенде, обеспечивает наглядную обратную связь при изучении программирования аппаратуры и может быть использовано в лабораторных работах по цифровой схемотехнике и проектированию встраиваемых систем. Результаты работы полезны студентам, преподавателям и инженерам в области электроники.

Ключевые слова: программирование, интегральная схема, учебный стенд, Cyclone, Intel/Altera, Verilog

DISPLAY INTERFACE FOR TRAINING STAND BASED ON A PROGRAMMABLE LOGIC INTEGRATED CIRCUIT

Boldyrev A.V.

*Federal State Budgetary Educational Institution of Higher Education
"Don State Technical University Rostov-on-Don", Rostov-on-Don,
e-mail: avb49@aaanet.ru*

The article addresses the relevant task of teaching students in technical specialties the programming of electronic devices based on the hardware description language Verilog. The aim of the study is to develop and verify an educational project using the example of a display interface as part of a training stand. Within the scope of the research, an analysis was conducted of the architecture of the training stand and the data exchange protocol between the programmable logic integrated circuit and the display. Special attention is given to justifying the use of a 4-bit data transmission mode, which allows optimizing the utilization of the chip's pins. The project was implemented in Verilog. Modeling and debugging were performed in the Quartus II environment, and a programmable logic integrated circuit of the training stand was used for experimental verification. The developed driver module initializes the display; converts 8-bit data into a sequence of nibbles for transmission over a 4-bit bus and controls the main display signals. For testing purposes, a module was created that generates 8-bit character codes. Experimental verification of the driver module confirmed the correct operation of the interface. The resulting solution simplifies the debugging and modeling of circuits on the training stand, provides visual feedback when learning hardware programming, and can be used in laboratory works on digital circuit design and embedded systems engineering. The results of the work are valuable for students, educators, and engineers in the electronics field.

Keywords: programming, integrated circuit, training stand, Cyclone, Intel/Altera, Verilog

Введение

Изучение языков программирования аппаратуры в колледжах имеет ряд важных преимуществ, связанных с развитием технических навыков, подготовкой к будущей профессии и пониманием принципов работы современных цифровых систем [1, 2]. Эти языки являются ключевыми инструментами в области проектирования цифровых устройств, включая программируемые ло-

гические интегральные схемы (ПЛИС) [3]. Несмотря на отличие от традиционных языков программирования, они развивают алгоритмическое мышление и умение структурировать задачи, требуют чёткого описания поведения системы, поддерживают моделирование и проверку проектов до их физической реализации. Это способствует развитию у студентов умений анализировать и отлаживать системы на ранних

этапах разработки, снижая риск ошибок. Знание этих языков повышает конкурентоспособность выпускников на рынке труда [4], а также может стать отправной точкой для освоения более сложных инструментов или языков высокого уровня для системного проектирования [5].

Наряду с этим все более важным становится понимание принципов работы ПЛИС, получивших широкое применение в промышленности, особенно в сферах телекоммуникаций, аэрокосмической отрасли, потребительской электроники и разработки микросхем [6]. Особое значение понимание этих принципов приобретает в эпоху развития интернета вещей и искусственного интеллекта [7].

Изучение ПЛИС и языков программирования аппаратуры объединяет знания из математики, физики, информатики и инженерии, способствуя комплексному пониманию технологий. При этом работа над реальными проектами (например, создание простых цифровых устройств) мотивирует учащихся и развивает навыки командной работы [8, с. 18].

Таким образом, введение языков программирования ПЛИС в программу колледжей способствует формированию инженерных компетенций, пониманию основ современной электроники и подготовке к будущим профессиональным вызовам. Однако важно учитывать, что эти языки требуют тщательного изучения и практики, а их преподавание должно сопровождаться доступом к соответствующим инструментам.

Цель исследования – разработка и верификация образовательного проекта на примере интерфейса дисплея в составе учебного стенда для обучения студентов технических колледжей основам языков программирования ПЛИС.

Материалы и методы исследования

Аппаратной платформой исследования является учебный стенд SDK-6.1/E производства ООО «ЛМТ» (г. Санкт-Петербург) [9, 10, 11]. Стенд предназначен для изучения основ проектирования современных электронных модулей на базе ПЛИС средней степени интеграции. Наличие в составе стенда ПЛИС Cyclone производства Intel/Altera¹ и подсистемы памяти (FLASH, SRAM, EEPROM) позволяет создавать на его основе полнофункциональное вычислительное ядро. Роль буквенно-цифрового знак синтезирующего модуля в стенде выполняет жидкокристаллический (ЖК) дисплей модели WH1602B-YGK-CP.

¹ Cyclone FPGA Family [Электронный ресурс]. Режим доступа: https://www.chipfind.ru/datasheet/pdf/_altera/ep1c6.pdf (дата обращения 22.11.2025).

Дисплей имеет внутреннюю память для определения матрицы символов, а также стандартного множества символов.

Программными средствами исследования являются среда проектирования Quartus II компании Intel/Altera² и язык описания аппаратуры Verilog [12, 13].

Исходными материалами исследования являются также руководство пользователя стенда SDK-6.1/E, спецификации ЖК дисплея WH1602B YGK CP³ и контроллера⁴.

Анализ архитектуры стенда и протокола обмена данными между ПЛИС и ЖК дисплеем определил необходимость исследования особенностей 4-разрядного режима работы дисплея. Одной из проблем, возникающих в процессе отладки и моделирования схем на стенде SDK-6.1/E [14], является построение интерфейса управления ЖК дисплеем, выводы которого соединены с ПЛИС в соответствии с таблицей.

Младшие выводы данных (DB0 – DB3) дисплея не используются, т. к. в стенде реализуется 4-разрядный интерфейс.

На основе результатов анализа проведено проектирование интерфейса: разработаны схема драйвера дисплея и модуль ее тестирования на языке Verilog, реализованы алгоритмы инициализации и передачи данных.

Для моделирования и отладки драйвера и тестера дисплея использовалась среда Quartus II, в которой проводилась проверка временных диаграмм и верификация преобразования 8-битных данных в две 4-битные посылки.

Для подтверждения работоспособности предложенного решения была проведена экспериментальная проверка путем загрузки разработанных модулей образовательного проекта в ПЛИС стенда SDK-6.1/E.

Результаты исследования и их обсуждение

При выполнении образовательного проекта разработан интерфейс управления дисплеем для учебного стенда SDK 6.1/E на базе ПЛИС Cyclone компании Intel/Altera, реализующий 4 разрядный режим обмена данными.

Это позволило эффективно использовать ограниченные ресурсы стенда при сохранении функциональности отображения информации.

² Quartus II Web Edition (Win) [Электронный ресурс]. Режим доступа: <https://soltau.ru/index.php/plis/item/360-install-quartus/> (дата обращения 22.11.2025).

³ LCD 1602B Winstar [Электронный ресурс]. Режим доступа: <https://xchem.net/mc/mc89.php/> (дата обращения: 22.11.2025).

⁴ Подключение и использование дисплея на базе HD44780 [Электронный ресурс]. Режим доступа: <https://microtechnics.ru/podklyuchenie-i-rabota-s-displeem-wh1602/> (дата обращения: 22.11.2025).

Соединения ЖК дисплея и ПЛИС в стенде SDK-6.1/E

Функции дисплея	Вывод дисплея	Вывод ПЛИС
RS (выбор регистра)	4	132
R/W (чтение/запись)	5	130
E (сигнал разрешения)	6	131
DB4 (данные)	11	124
DB5 (данные)	12	125
DB6 (данные)	13	128
DB7 (данные)	14	129

Примечание: составлена автором на основе полученных данных в ходе исследования.

Разработанный интерфейс устраняет одну из проблем отладки схем на стенде – сложность интеграции дисплея с ПЛИС. Использование 4 разрядного режима оптимизирует количество задействованных выводов ПЛИС (всего 7 сигналов вместо 11 при 8 разрядном интерфейсе).

В результате анализа спецификации ЖК дисплея WH1602B YGK CP определены ключевые этапы передачи данных между ПЛИС и дисплеем.

До того, как ЖК дисплей сможет отобразить записанную в него информацию, он должен быть проинициализирован и запрограммирован. Программирование выполняется путем удержания входов RS и R/W в низком состоянии, подачи команд на входы с DB7 по DB0 и выдачи строб-сигнала E.

После инициализации ЖК дисплея в него записываются 8-битные данные кода ASCII путем установки входов RS и R/W в 1 и 0 соответственно, помещения данных кода ASCII на входы DB7 – DB0, формирования сигнала E и предоставления порядка 40 микросекунд времени для выполнения операции. Каждый раз, когда байт данных записывается в ЖК дисплей, курсор перемещается на одну позицию вправо.

При организации 4-разрядного интерфейса байт передается в 2 посылки, старшим битом вперёд. В спецификации дисплея первая посылка обозначается D7 (старшая тетрада), а вторая – D3 (младшая тетрада). Перед следующей посылкой проверяется наличие флага занятости. Если он отсутствует, то процесс передачи можно продолжить, в противном случае придется подождать, пока контроллер, управляющий дисплеем, завершит предыдущие операции. Для упрощения инициализации чтение флага занятости может быть заменено задержкой. В соответствии с этим для передачи

байта в 2 посылки необходимо выполнить следующие шаги:

- убедиться, что управляющий контроллер свободен,
- установить RS в 0 (команда) или 1 (данные), в зависимости оттого, что передается,
- R/W установить в 0,
- установить строб-сигнал E в 1,
- поместить на шину данных (DB4 – DB7) старшую тетраду D7 передаваемого байта,
- выполнить задержку 2мкс,
- сбросить строб-сигнал E в 0,
- выполнить задержку 1мкс,
- установить строб-сигнал E в 1,
- поместить на шину данных (DB4 – DB7) младшую тетраду D3 передаваемого байта,
- выполнить задержку 2мкс,
- сбросить строб-сигнал E в 0.

Схема интерфейса управления дисплеем реализована на языке программирования аппаратуры Verilog [15]. Оформленный в виде модуля драйвера LCD_drv_8_to_4 в пакете Quartus II, интерфейс осуществляет инициализацию дисплея и преобразование 8-битных данных в последовательность полубайтов, позволяющую подключиться к дисплею по 4-разряднойшине. Функционирование схемы поясняют комментарии в коде описания модуля, что важно при выполнении образовательного проекта на Verilog.

Общая схема тестирования драйвера ЖК дисплея применительно к ПЛИС стенда SDK-6.1/E показана на рисунке. Генерируемые тестером char_gen символы выводятся с помощью драйвера LCD_drv_8_to_4 на дисплей. Управляющий сигнал схемы reset формируется на плате стенда переключателем SW5, а сигналы тактовой последовательности clk задаются внутренней системой синхронизации стенда.

```

module LCD_drv_8_to_4 (
    // системные сигналы
    input r,           // сброс системы
    input c,           // синхронизация
    // входные порты
    input [7:0] di,     // 8-битные ASCII
    input de,
    // выходные порты LCD 4-bit
    output [3:0] lcd_do,   // LCD 4-битные данные
    output lcd_e,       // LCD E строб
    output lcd_rs,      // LCD R/S: 0 – команда, 1 – данные
    output lcd_rw,      // LCD R/W: 0 – запись, 1 – чтение
    output busy );
`define ms_clk_val 50000 // такты на 1 мс; 50000 – для частоты @ 50 МГц, 500 – для отладки/
моделирования
`define lcd_E_duration 30 // 600 нс @ 50 МГц (600 * 50 / 1000)
reg [7:0] di_r;
reg [2:0] state; // рабочее состояние
reg [3:0] i_state; // состояние инициализации ЖК дисплея
reg [4:0] cnt;
    reg [15:0] init_seq [10:0]; // начальная последовательность
reg [15:0] cnt_1ms;
reg [5:0] delay_ms;
    wire next_stt_en;
    wire [2:0] next_stt_val;
    wire cnt_en;
        wire de4, de8; // 1 или 2 срезающих строба
    wire is_init = i_state != 10; // инициализация в процессе
    wire de_i; // data enable internal strobe
    wire in_transmitt = (state != 0);
    wire [15:0] d_init;
    wire next_init_en;
    wire cnt_1ms_zero = cnt_1ms == 0;
    wire cnt_1ms_en = 1;
    wire delay_ms_zero = delay_ms == 0;
    assign d_init = init_seq[i_state];
    assign busy = in_transmitt | is_init;
    assign lcd_do = ((state == 1) | (state == 2)) ? di_r[7:4] : di_r[3:0];
    assign next_stt_en = in_transmitt ? cnt == 0 : de_i;
    assign cnt_en = in_transmitt;
    assign lcd_e = ((state == 1) | (state == 3));
    assign lcd_rs = !is_init;
    assign lcd_rw = 0;
    assign next_stt_val = (state == 4) ? 0 : (state + 1);
    assign next_init_en = is_init & !in_transmitt & delay_ms_zero;
    assign de4 = !d_init[8] & next_init_en;
    assign de8 = d_init[8] & next_init_en;
    assign de_i = (de & !busy) | (d_init[9] & (de8 | de4));
task reset_reg;
begin
    cnt <= 0;
    state <= 0;
    di_r <= 0;
    i_state <= 0;
    delay_ms <= 0;
    cnt_1ms <= 0;
end
endtask
initial begin
    // инициализация всех регистров
    reset_reg();

```

```

    init_seq[0] <= 16'b_110010_0_0_00000000; // ожидание 50 мс после включения
init_seq[1] <= 16'b_000101_1_0_00000011; // установить 8-битный режим
init_seq[2] <= 16'b_000101_1_0_00000011; // установить 8-битный режим
init_seq[3] <= 16'b_000101_1_0_00000011; // установить 8-битный режим
init_seq[4] <= 16'b_000101_1_0_00000010; // установить 4-битный режим
init_seq[5] <= 16'b_000001_1_1_00101000; // 28 2 строки, шрифт char
init_seq[6] <= 16'b_000001_1_1_00001100; // 0C дисплей включить
init_seq[7] <= 16'b_000011_1_1_00000001; // 01 очистка дисплея
init_seq[8] <= 16'b_000001_1_1_10000000; // 80 курсор влево/вверх по экрану
init_seq[9] <= 16'b_000000_0_0_00000000; // конечное состояние – без инициализации
init_seq[10] <= 16'b_000000_0_0_00000000; // конечное состояние – без инициализации
end
always@(posedge c) begin
if(r) begin
  reset_reg();
end else begin
  if(next_stt_en) begin
    case(state)
      0: if(de_i) state <= de4 ? 3 : 1;
      4: state <= 0;
      default: state <= state + 1;
    endcase
  end
  if(next_init_en) begin
    i_state <= i_state + 1;
    delay_ms <= d_init[15:10];
  end else begin
    delay_ms <= delay_ms - cnt_1ms_zero;
  end
  cnt_1ms <= (next_init_en | cnt_1ms_zero) ? `ms_clk_val : cnt_1ms - !delay_ms_zero; // счетчик на 1 мс
    if(de_i) begin
      di_r <= is_init ? d_init[7:0] : di;
      cnt <= `lcd_E_duration;
    end else begin
      cnt <= cnt - cnt_en;
    end
  end
end
endmodule

```

Для тестирования драйвера дисплея на языке Verilog создан также модуль `char_gen`, генерирующий 8 разрядные коды символов ASCII.

```

module char_gen (
  // системные сигналы
  input c,           // синхронизация
  // входные данные
  output [7:0] dout, // 8-битный ASCII
  output oe
);
reg [7:0] d_r;
reg [23:0] cnt;
  wire inc = cnt == 0;
  assign oe = inc;
  assign dout = d_r;
initial begin
  cnt <= 0;
  d_r <= 48;
end
always@(posedge c) begin
  cnt <= cnt + 1;
  if(inc) begin
    d_r <= (d_r == 122) ? 48 : d_r + 1;
  end
end
endmodule

```

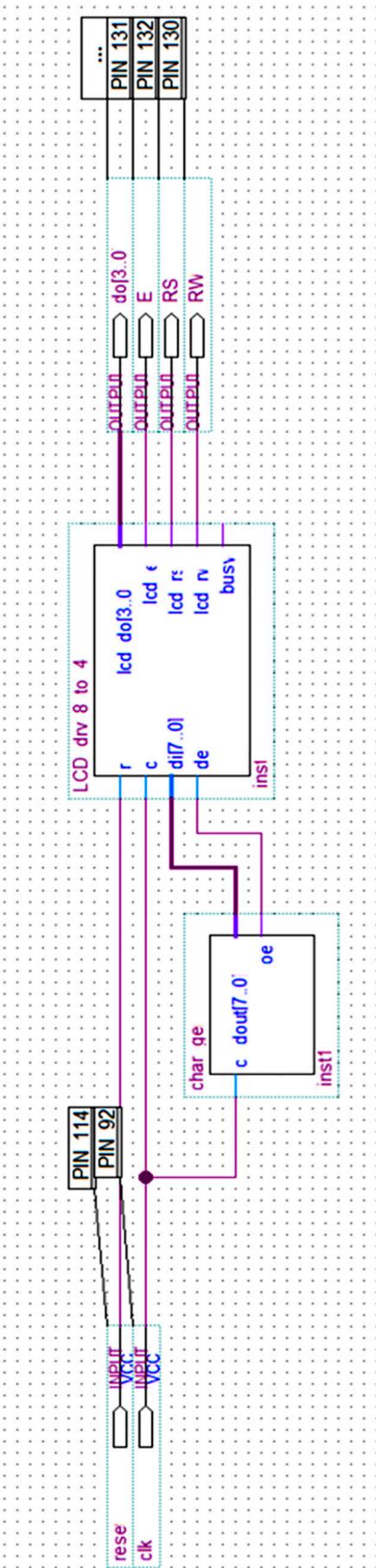


Схема тестирования драйвера ЖК дисплея для стенда SDK-6.1/E
Примечание: составлена автором по результатам данного исследования

Результаты тестирования подтверждают корректную работу 4-разрядного интерфейса и эффективность архитектуры модуля драйвера в части ее универсальности (подходит для дисплеев на базе контроллера HD44780) и масштабируемости (возможность адаптации под другие разрешения ЖК дисплеев).

Практическая значимость исследования заключается в том, что разработанное решение упрощает процесс отладки и моделирования схем на учебном стенде, обеспечивает наглядную обратную связь через дисплей при изучении программирования ПЛИС, а также может быть использовано в лабораторных работах по цифровой схемотехнике и проектированию встраиваемых систем.

Заключение

В ходе проведенного исследования решена задача разработки и верификации образовательного проекта для обучения студентов технических колледжей основам языков программирования ПЛИС.

Исследование проводилось на примере решения проблемы интеграции ЖК дисплея модели WH1602B YGK CP со стеном SDK 6.1/E. В результате обоснована целесообразность применения 4-разрядного режима обмена данными, позволяющего оптимизировать использование выводов ПЛИС.

Разработан модуль драйвера ЖК дисплея на языке Verilog, который выполняет инициализацию дисплея согласно требованиям контроллера HD44780, преобразует 8-битные данные в последовательность полубайтов для передачи по 4-разряднойшине, управляет основными сигналами дисплея в соответствии с протоколом обмена.

Создан тестовый модуль char_gen, генерирующий 8-разрядные коды символов ASCII для проверки работоспособности интерфейса.

Экспериментально подтверждена корректность функционирования разработанного решения.

Результаты работы представляют интерес для студентов, изучающих проектирование на ПЛИС, преподавателей, разрабатывающих лабораторные работы и инженеров, работающих с интерфейсами ЖК дисплеев.

Список литературы

1. Алексеевский П.И. Обучение студентов архитектуре вычислительных систем с использованием языка Verilog // Педагогическое образование в России. 2016. № 7. С. 131-138. URL: <https://pedobrazovanie.ru/archive/345568/7/obuchenie-studentov-arkhitekture-vychislitel-nykh-sistem-s-ispolzovaniem-yazyka-verilog> (дата обращения: 22.11.2025).

2. Mayoz, C.A., da Silva Beraldo A.L., Villar-Martinez A., Rodriguez-Gil L., de Souza Seron W.F.M., Orduña P. FPGA remote laboratory: experience of a shared laboratory between UPNA and UNIFESP // 2020 XIV technologies applied to electronics teaching conference (TAE). 2020. Р. 1-8. URL: https://www.academia.edu/44036178/FPGA_remote_laboratory_experience_of_a_shared_laboratory_between_UPNA_and_UNIFESP (дата обращения: 22.11.2025).

3. Басс А.В., Антонов М.А. Работа с ПЛИС с использованием языка описания аппаратуры Verilog // Известия ТулГУ. Технические науки. 2019. Вып. 3. С. 19-24. URL: <https://elibrary.ru/item.asp?id=38187279> (дата обращения: 22.11.2025).

4. Белащ О.Ю., Чиркова А.А. Оценка востребованности молодых специалистов на основе

опроса работодателей // Инновации. 2018. № 9. С. 86-89. URL: <https://cyberleninka.ru/article/n/otsenka-vostrebovannosti-molodyh-spetsialistov-na-osnove-oprosa-rabotodateley/viewer> (дата обращения: 22.11.2025).

5. Лоренс А.С., Петухов Г.А. Разработка стенда для удаленной асинхронной работы с ПЛИС // Маркуковские научные чтения. 2022. Т. 1. С. 120. URL: <https://cyberleninka.ru/article/n/otobrazhenie-neskladiruemyh-resursov-v-modelyah-kalendarnogo-planirovaniya-proektov/pdf> (дата обращения: 22.11.2025). DOI: 10.24412/cl-35065-2022-1-01-71.

6. Boutros A., Betz V. FPGA architecture: Principles and progression // IEEE Circuits and Systems Magazine. 2021. Vol. 21. № 2. Р. 4-29. URL: https://andrewboutros.github.io/assets/pdf/papers/014_casm2021_architecture.pdf (дата обращения: 22.11.2025).

7. Nechi A., Groth L., Mulhem S., Merchant F., Buchty R., Berekovic M. Fpga-based deep learning inference accelerators: Where are we standing? // ACM Transactions on Reconfigurable Technology and Systems. 2023. Vol. 16. № 4. Р. 1-32. URL: <https://dl.acm.org/doi/pdf/10.1145/3613963> (дата обращения: 22.11.2025).

8. Максфилд К. Проектирование на ПЛИС. Архитектура, средства и методы. М.: ДМК Пресс, 2016. 408 с. ISBN: 978-5-97060-265-2.

9. Степаненко Д.Р., Болдырев А.В. Обзор учебных стендов для обучения программированию ПЛИС // Инновации и инжиниринг в формировании инвестиционной привлекательности региона: сборник научных трудов II открытого международного научно-практического форума. Ростов-на-Дону: Издательство ДГТУ-ПРИНТ, 2017. С. 446-451. URL: <https://elibrary.ru/item.asp?id=32329091> (дата обращения: 22.11.2025).

10. Ключев А.О., Плетунов А.Е., Дергачев А.М. Опыт использования лабораторных стендов SDK в учебном процессе // Научно-технический вестник информационных технологий, механики и оптики. 2019. Т. 19. № 1. С. 184-187. URL: <https://ntv.ifmo.ru/file/article/18451.pdf> (дата обращения: 22.11.2025). DOI: 10.17586/2226-1494-2019-19-1-184-187.

11. Можный А.Д., Болдырев А.В. Применение стендов SDK-6.1 для отладки проектов в Altium Designer // Форум молодых ученых. 2017. № 5. С. 1439-1450. EDN: ZBNRAX.

12. Емец С. Verilog – инструмент разработки цифровых электронных схем // Компоненты и технологии. 2001. № 4. С. 76-79. URL: <https://elibrary.ru/item.asp?edn=mtxawx> (дата обращения: 22.11.2025).

13. Salcic Z., Smailagic A. Digital Systems Design and Prototyping: Using Field Programmable Logic and Hardware Description Languages. Springer Publishing Company, Incorporated, 2013. 621 p. ISBN: 978-1-4757-8419-0.

14. Еремин В.В., Мальчуков А.Н. О применении блочно-ориентированного подхода к разработке устройств на ПЛИС // Вестник науки Сибири. 2011. № 1. С. 379-381. URL: <https://earchive.tpu.ru/bitstream/11683/14958/1/132.pdf> (дата обращения: 22.11.2025).

15. Соловьев В.В. Основы языка проектирования цифровой аппаратуры Verilog. М.: Горячая линия-Телеком, 2024. 284 с. ISBN: 978-5-9912-0923-6.

Конфликт интересов: Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest: The authors declare that there is no conflict of interest.