

*Болдырев А.В.,  
кандидат технических наук,  
доцент кафедры «Автоматизация производственных процессов»  
Донской Государственный Технический Университет  
Россия, г. Ростов-на-Дону*

## **ДРАЙВЕР ЖИДКО-КРИСТАЛЛИЧЕСКОГО ИНДИКАТОРА НА ОСНОВЕ ПЛИС**

***Аннотация:** В данной статье рассматривается одна из проблем, возникающих в процессе отладки и моделирования схем на стенде, предназначенном для обучения программированию ПЛИС. Представлены особенности коммуникации жидкокристаллического индикатора (ЖКИ) в стенде, коды описания на языке Verilog схем драйвера ЖКИ и модуля его тестирования. Полученные результаты подтверждают корректную работу драйвера.*

***Ключевые слова:** ПЛИС; программирование; ЖКИ; Cyclone; Intel/Altera; Verilog; ASCII.*

## **PLD-BASED LIQUID CRYSTAL DISPLAY DRIVER**

***Abstract:** This article addresses one of the issues arising during debugging and simulation of circuits on a stand designed for PLD programming training. It presents specifics of liquid crystal display (LCD) communication in the stand, as well as code descriptions in Verilog language for both the LCD driver circuit and its test module. The results obtained confirm correct operation of the driver.*

***Keywords:** PLD; programming; LCD; Cyclone; Intel/Altera; Verilog; ASCII.*

Во многих вузах РФ успешно используется учебный стенд SDK-6.1/E производства ООО «ЛМТ» (г. Санкт-Петербург) [1]. Стенд предназначен для изучения основ проектирования современных электронных модулей на базе ПЛИС средней степени интеграции. Наличие в составе стенда ПЛИС CYCLONE компании Intel/Altera и подсистемы памяти (FLASH, SRAM, EEPROM) позволяет создавать на его основе полнофункциональное вычислительное ядро. Роль буквенно-цифрового знак синтезирующего модуля в стенде выполняет жидкокристаллический индикатор модели WH1602B-YGK-CP с разрешением 16 символов x 2 строки [2, 3]. Дисплей имеет внутреннюю память для определения матрицы символов, а также стандартного множества символов.

Одной из проблем, возникающих в процессе отладки и моделирования схем на стенде SDK-6.1/E, является построение драйвера управления ЖКИ, выходы которого соединены с ПЛИС в соответствии с таблицей 1. Младшие выходы данных (DB0 – DB3) дисплея не используются, т. к. в стенде реализуется 4-разрядная шина [4].

*Таблица 1.*

*Соединения ЖКИ 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

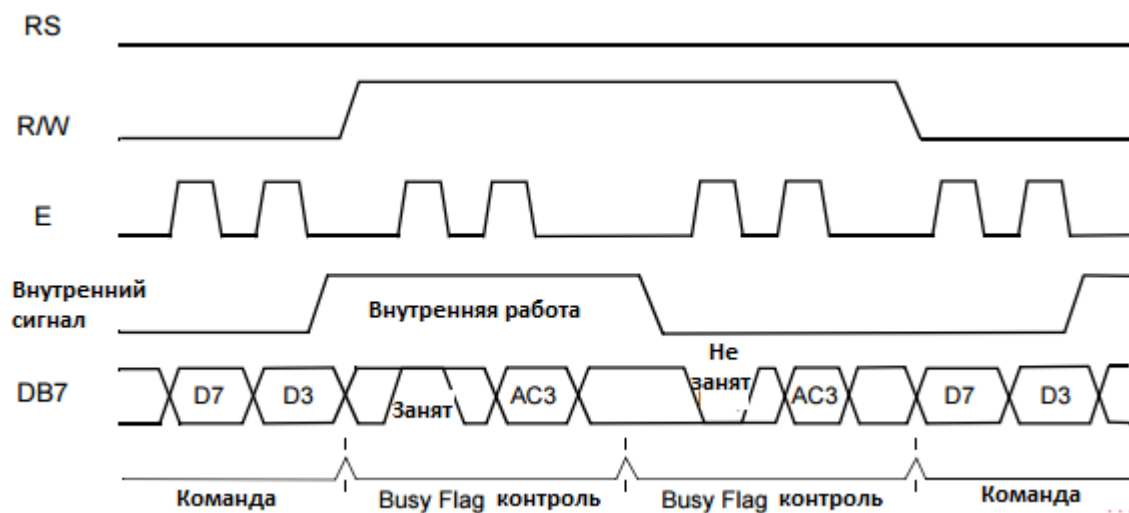
До того, как ЖКИ сможет отобразить записанную в него информацию, он должен быть проинициализирован и запрограммирован [5]. Программирование выполняется путем удержания входов RS и R/W в низком состоянии, подачи команд на входы с DB7 по DB0 и выдачи строб-сигнала E.

После инициализации ЖКИ 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 ЖКИ, курсор перемещается на одну позицию вправо. Описанные процедуры иллюстрирует рисунок 1.



**Рисунок 1. Временная диаграмма передачи данных и команд**

При организации 4-разрядного интерфейса байт передается в 2 посылки, старшим битом вперед. На рисунке 2 первая посылка обозначена D7 (старшая тетрада), а вторая – D3 (младшая тетрада). Перед следующей посылкой проверяется наличие флага занятости Busy Flag. Если он отсутствует, то процесс передачи можно продолжить, в противном случае придется подождать, пока контроллер, управляющий ЖКИ, завершит предыдущие операции. Для упрощения инициализации чтение флага занятости может быть заменено задержкой.



*Рисунок 2. Временная диаграмма работы 4-разрядного интерфейса*

В соответствии с рисунком 2 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 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 передаваемого байта,
- выполнить задержку 2мкс,
- сбросить строб-сигнал E в 0.

```

module LCD_drv_8_to_4_2
(
  // 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 ЖКИ
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

```

### ***Рисунок 3. Описание 4-разрядного драйвера ЖКИ***

Код описания на языке Verilog схемы драйвера ЖКИ показан на рисунках 3 – 5. Оформленный в виде модуля драйвера LCD\_drv\_8\_to\_4 (рисунок 6) в пакете Quartus II, драйвер осуществляет инициализацию дисплея и преобразование 8-битных данных в последовательность полубайтов, позволяющую подключиться к ЖКИ по 4-разрядной шине.

```

wire de_i; // данные разрешают внутреннее стробирование
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

```

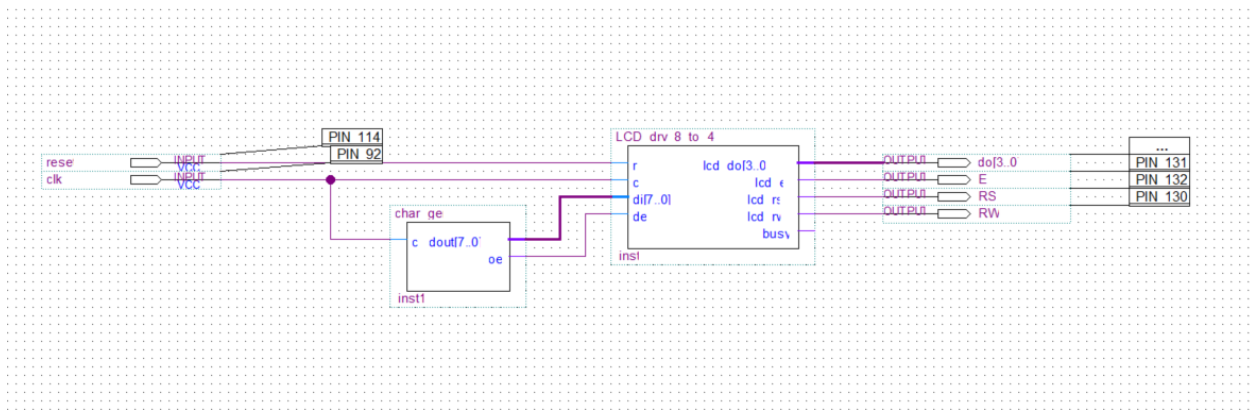
#### ***Рисунок 4. Продолжение описания 4-разрядного драйвера ЖКИ***

Для тестирования драйвера ЖКИ на языке Verilog был создан модуль счетчика char\_gen, код описания которого приведен на рисунке 7. Общая схема тестирования драйвера ЖКИ применительно к ПЛИС стенда SDK-6.1/Е показана на рисунке 6. Модуль счетчика генерирует 8-разрядные коды символов, которые с помощью драйвера LCD\_drv\_8\_to\_4 выводятся на дисплей. Управляющий сигнал схемы reset формируется на плате стенда DIP-

переключателем SW5.

```
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; // 0С 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
MC
    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
```

**Рисунок 5. Окончание описания 4-разрядного драйвера ЖКИ**



**Рисунок 6. Схема тестирования драйвера ЖКИ для стенда SDK-6.1/E**

```

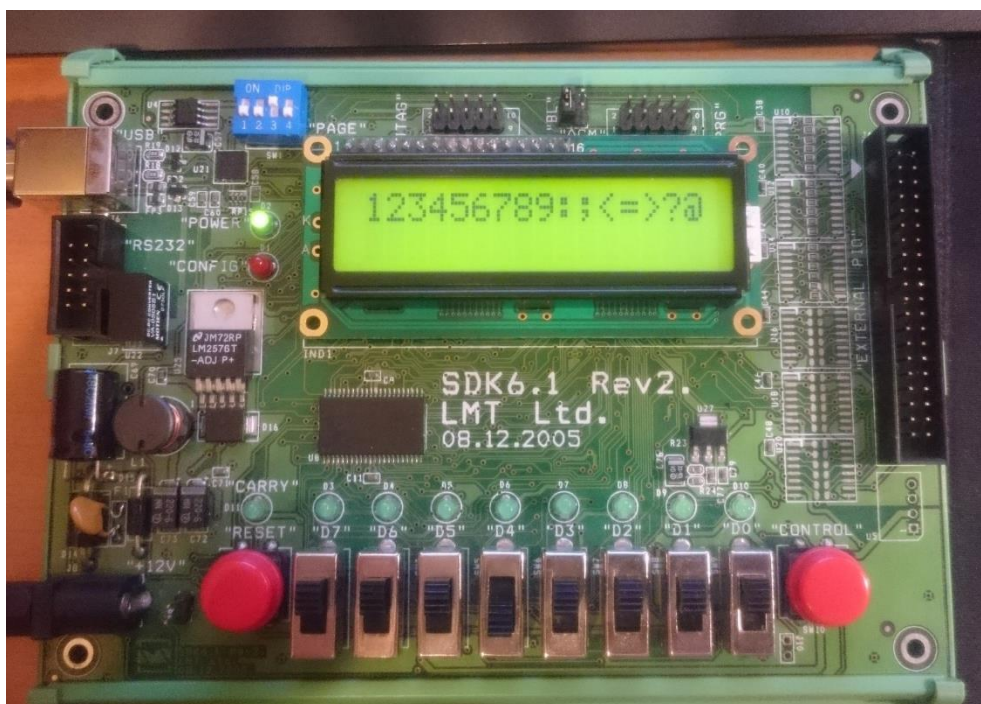
module char_gen
(
  // системные сигналы
  input c,           // синхронизация
  // входные данные
  output [7:0] dout, // 8-битный ASCII
  output oe
);
reg [7:0] d_r;
reg [23:0] cnt;

    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

```

**Рисунок 7. Описание тестера драйвера ЖКИ**



**Рисунок 8. ЖКИ стэнда SDK-6.1/E по окончании тестирования**

Результаты тестирования демонстрирует рисунок 8, показывающий вывод на ЖКИ в первой строке 16 символов кодов ASCII, что подтверждает корректную работу 4-разрядного драйвера.

#### **Используемые источники:**

1. Комплекс учебный лабораторный SDK-6.1. Руководство пользователя. – СПб.: ООО «ЛМТ», 2006.
2. Подключение и использование дисплея на базе HD44780 [Электронный ресурс]: URL: <https://microtechnics.ru/podklyuchenie-i-rabota-s-displeem-wh1602/> (дата обращения: 5.04.2021).
3. LCD 1602В компании Winstar [Электронный ресурс]: URL: <https://cxem.net/mc/mc89.php/> (дата обращения: 5.04.2021).
4. Interfacing LCD in 4-bit mode with 8051 [Электронный ресурс]: URL: [https://explorembedded.com/wiki/Interfacing\\_LCD\\_in\\_4-bit\\_mode\\_with\\_8051/](https://explorembedded.com/wiki/Interfacing_LCD_in_4-bit_mode_with_8051/) (дата обращения: 11.04.2021).

5. Наваби З. Проектирование встраиваемых систем на ПЛИС / пер. с англ. Соловьева В. В. – М.: ДМК Пресс, 2016.