

РАЗРАБОТКА СИГНАЛИЗАЦИИ НА ПРОГРАММИРУЕМОЙ ЛОГИЧЕСКОЙ ИНТЕГРАЛЬНОЙ СХЕМЕ

Шелле Всеволод Сергеевич

Смирнов Дмитрий Андреевич

студент

Омский государственный технический университет, г. Омск

Аннотация. В современном мире зачастую используется такое средство контроля, как сигнализация, которое сообщает субъектам об ошибке системы. В статье рассмотрена технология разработки сигнализации на программируемой логической интегральной схеме, показывающая простоту использования данной системы.

Ключевые слова: программируемая логическая интегральная схема, сигнализация, устройство, соединение, порт.

Сигнализация- это единица информации, предназначенная для предупреждения субъекта и привлечения к его вниманию неисправностей технического процесса. Также она является видом автоматического контроля, используемым для подачи сигналов об опасности, отказе системы, угрозе или предстоящем событии. Рассмотрим, например, автомобильную сигнализацию.

Основные функции сигнализации направлены на отпугивание неквалифицированных взломщиков (звуковыми и световыми сигналами) и предупреждение владельца машины о попытке взлома или угона автомобиля. При постановке машины в «режим охраны» с брелка дистанционного управления включается блок управления автосигнализации, все датчики и блокировки. После этого автосигнализация реагирует на все попытки механического взлома замков или стекол, снятие колес, перемещение машины вручную или с помощью автокрана (эвакуатора).

Основной принцип работы сигнализации автомобиля основан на срабатывании электронных датчиков, посылающих сигнал о постороннем воздействии на машину в блок управления охранной системой. Блок управления, в свою очередь, включает звуковые (сирену), световые (мигание фар головного света и аварийных фонарей) тревожные сигналы и посылает радиосигнал на брелок владельца машины [1, с.212].

Основной целью данного исследования является разработка упрощенной примитивной модели сигнализации, которая позволит значительно понять, как работает ее механизм, Нам потребуется запрограммировать шлейфы прибора, включить их в общую сеть. При перерезании провода, на который подается единица, потухает лампочка и этот сигнал передается на ключ

или в охранную службу, смотря где использовать. Чаще всего такие системы используют в сигнализации автомобилей. Эту систему подключают к центральному замку [2, с. 197].
Проведем исследования в программе Quartus Prime:

- 1) Для создания нового проекта нужно в программе Quartus Prime выбрать:
File→new→new Quartus Project
- 2) Создаем новый проект. Для этого выбираем пункт меню
File→New Project Wizard
- 3) Появляется диалоговое окно мастера-помощника создания проекта. Нажимаем кнопку *Next*
- 4) Выбираем папку, где будет размещен проект. Я выбрал *d:/Work/Quartus/ISS-161*. И указываем название проекта *shel*. Нажимаем кнопку *Next*
- 5) В строчке *Name filter* вводим название *5CSEMA5F31C6*. Нажимаем кнопку *Next→Finish*
- 6) Чтобы создать новый файл, принадлежащий проекту, выбираем пункт меню *File→New*
- 7) Нам нужно создать файл, описывающий наше устройство. Это *Verilog HDL File*
- 8) Начинаем писать код для нашей сигнализации. Для этого нам нужно две ПЛИС. Первую мы программируем так, что при отсутствии сигнала лампочка потухает

```
1  module shel (clk, in, out);
2  input in;
3  output out;
4  input clk;
5  reg A;
6  always @(posedge clk)
7  begin
8  if (in==0) A=1;
9  else A=0;
10 end
11 assign out=A;
12 endmodule
```

Рисунок 1. Рабочее поле №1 Verilog HDL File

- 9) Вторую ПЛИС мы программируем так, чтобы на вход поступала единица

```
1  module verilog1(clk,out);
2  input clk;
3  output out;
4  reg B;
5  always @(posedge clk)
6  begin
7  B=1;
8  end
9  assign out=B;
10 endmodule
11
12
```

Рисунок 2. Рабочее поле №2 Verilog HDL File

- 10) Прежде чем делать симуляцию нужно откомпилировать проект. Выбираем пункты *Compile Design→Analys and Sintesys* двойным нажатием левой мыши мы сохраняем результат
- 11) В верхнем поле рабочего окна выбираем *Assigments→Pin Planner*

12) В появившемся окне в поле *location* задаем соединение портам

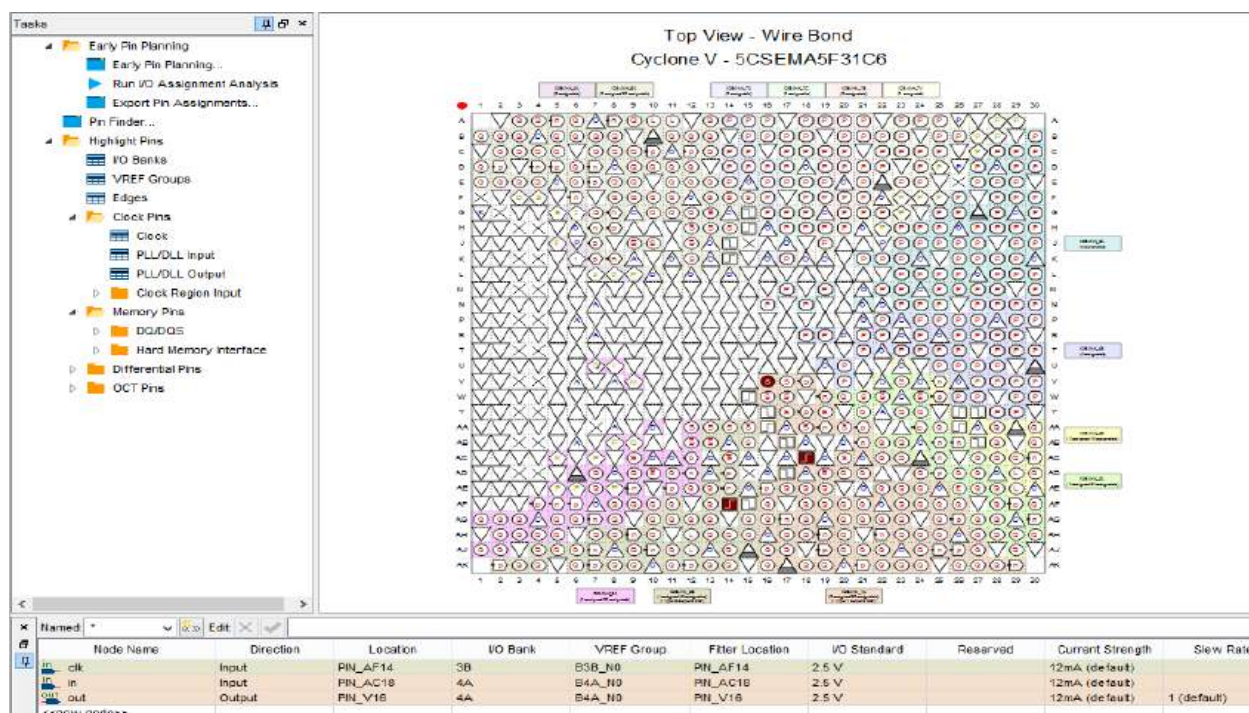


Рисунок 3. Рабочее окно Pin Planer в программе Quartus Prime для первой ПЛИС

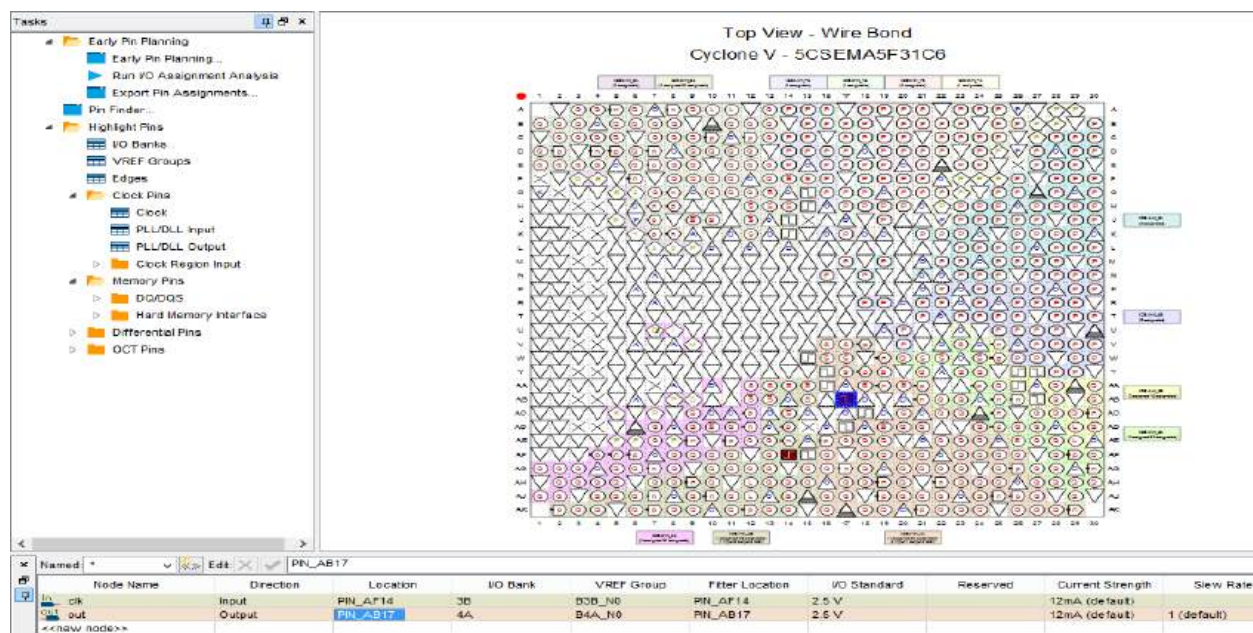


Рисунок 4. Рабочее окно Pin Planer в программе Quartus Prime для второй ПЛИС

- 13) Затем закрываем окно *Pin Planner*, нажимаем 2 раза на *Compile Design* → *Program Device*
- 14) Нажимаем кнопку *Hardware Setup* выбираем SOC и закрываем окно
- 15) Слева в окне нажимаем *Auto Detect* и выбираем 5CSEMA5F31C6
- 16) В окне появится два устройства, выбираем 5CSEMA5F31C6, нажимаем *Change File* для изменения прошивки
- 17) Выбираем *output files*, ставим галочку в окне *Program/Configure* и нажимаем *Start*

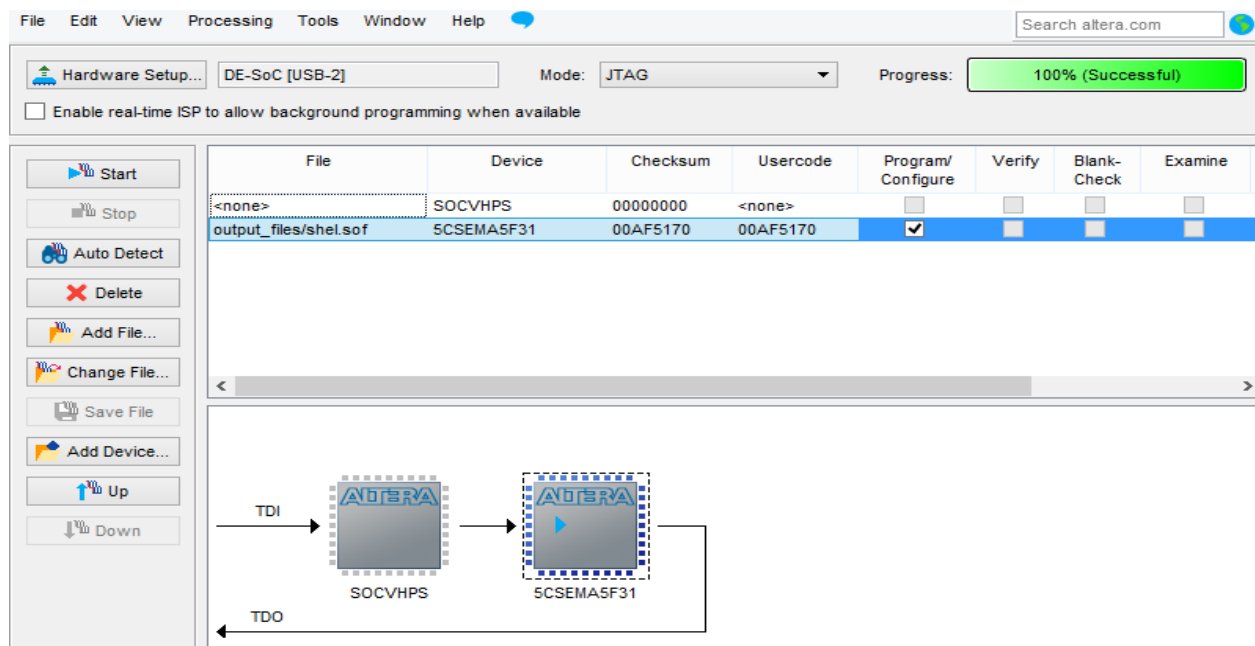


Рисунок 5. Рабочее окно Program Device для первой ПЛИС

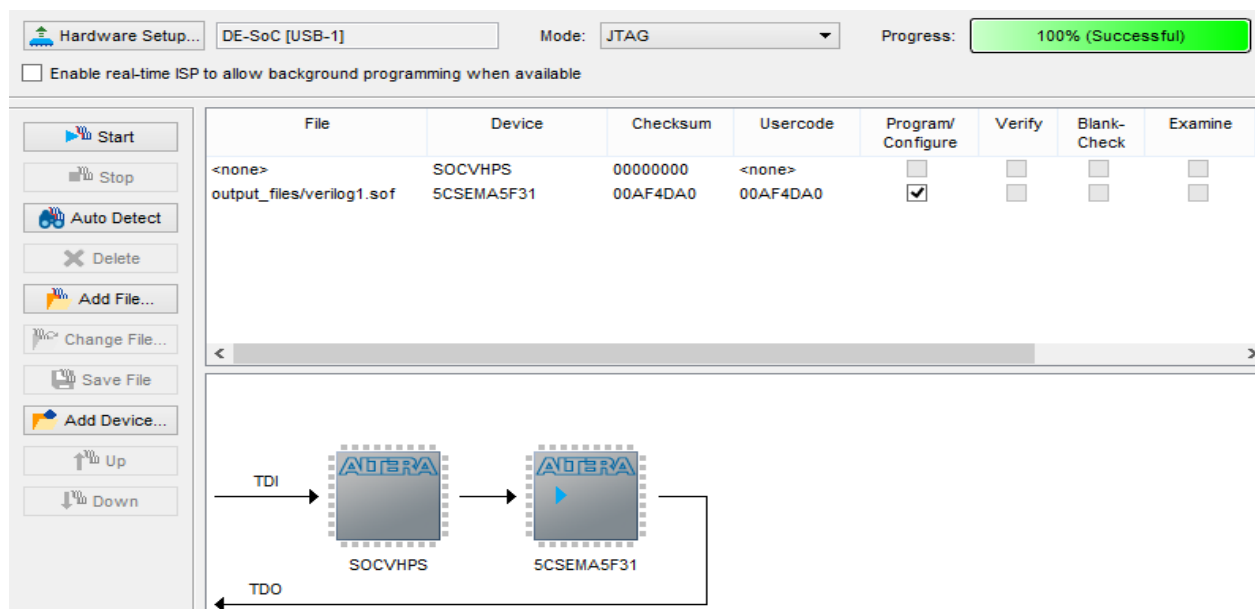


Рисунок 6. Рабочее окно Program Device для второй ПЛИС

18) Проверяем работу сигнализации

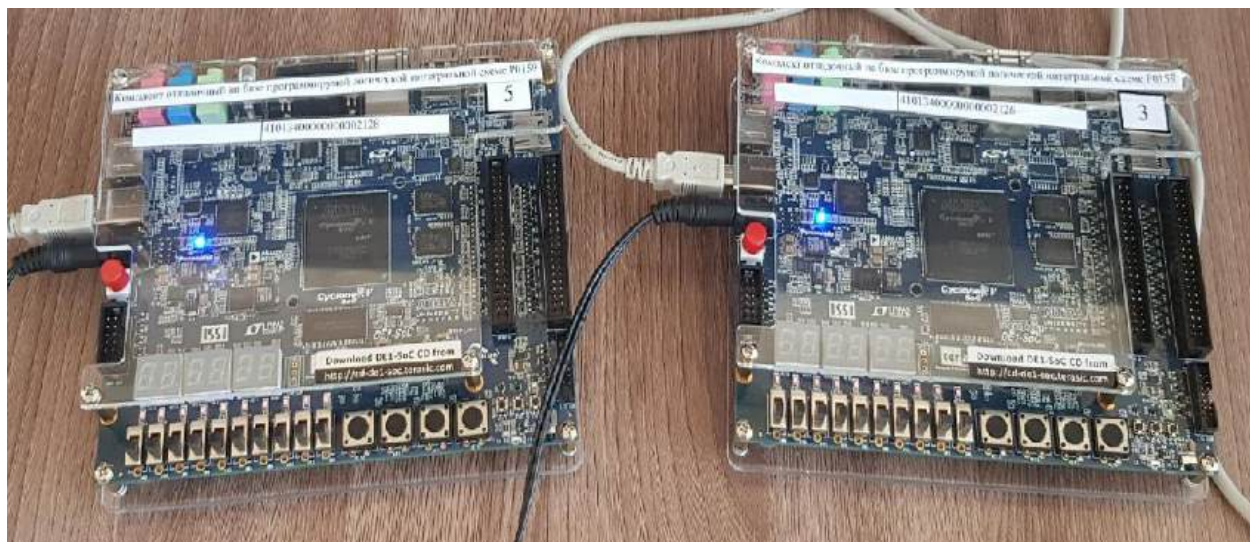


Рисунок 7. Разработанная сигнализация на ПЛИС в действии

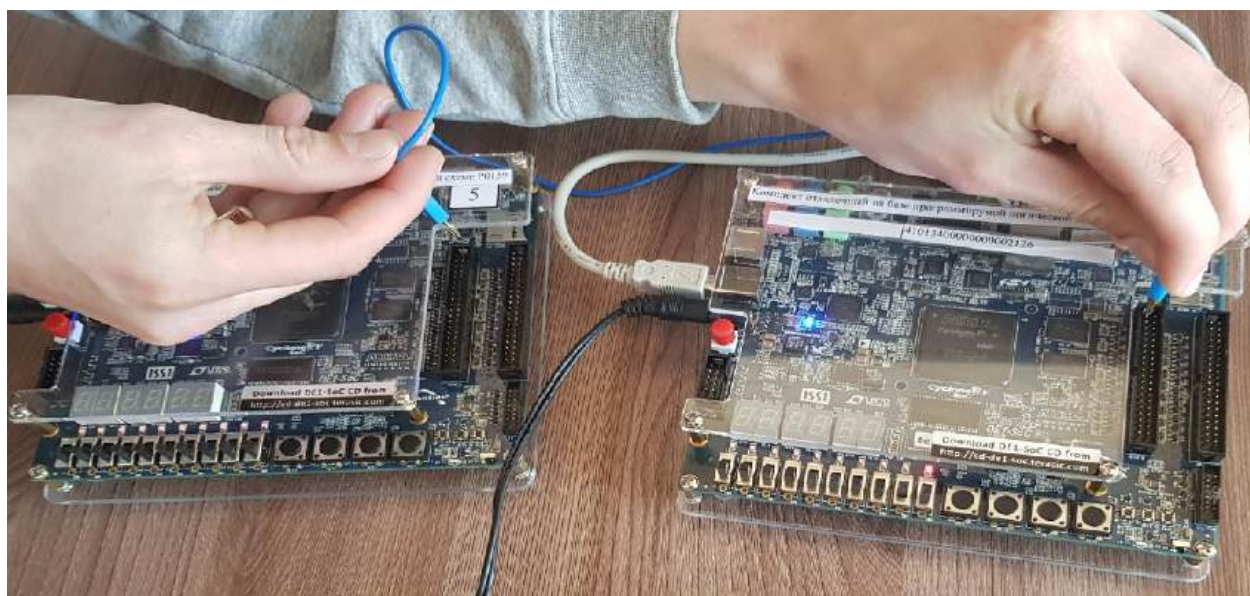


Рисунок 8. Разработанная сигнализация на ПЛИС в действии с помощью контактов

Список литературы:

1. Барсуков, В.С. Безопасность: технологии, средства, услуги / В.С. Барсуков. – М., 2001 – 496 с.
2. Ярочкин, В.И. Информационная безопасность. Учебник для студентов вузов / 3-е изд. – М.: Академический проект: Трикста, 2005. – 544 с.
3. DE1.SoC User Manual.

