Московский Авиационный институт

(национальный исследовательский университет)

Институт №7 «Робототехнические и интеллектуальные системы»

Кафедра 702 «Системы приводов авиационно-космической техники»

**Отчет по Лабораторной работе №4**

**по курсу «Аналоговые, дискретные и микропроцессорные устройства систем приводов»**

**Вариант 4**

Выполнил студент гр.7О-404С-17:

Ковальджи К. В.

Проверил д.т.н., профессор каф.702:

Кривилёв А.В.

Москва 2020

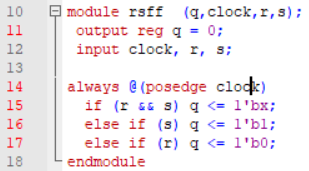
**Шестиразрядный суммирующий счетчик в коде Джонсона**

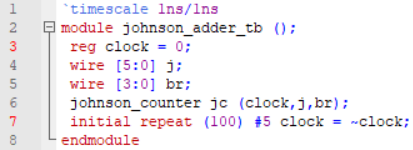
Для описания на Verilog HDL и отладки в ModelSim составим таблицу истинности. Для реализации счетчика в коде Джонсона был выбран RS-триггер. С целью получения необходимых для работы триггера информационных сигналов организуем таблицу истинности следующим образом:

* для удобства навигации в таблице запишем номера наборов в первый столбец;
* запишем текущее состояние и следующее состояние в следующие два больших столбца;
* для еще большего удобства работы с таблицей применим заливку различных оттенков таким образом, чтобы соответствующие разряды имели один и тот же цвет для всех больших столбцов;
* получим функции переходов для каждого перехода из текущего состояния в следующее;
* при помощи словаря переходов отыщем информационные сигналы, соответствующие полученным функциям переходов.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Текущее  состояние | | | | | | Следующее состояние | | | | | | Функции  переходов | | | | | | Информационные  сигналы | | | | | | | | | | | |
| № | J5 | J4 | J3 | J2 | J1 | J0 | J5 | J4 | J3 | J2 | J1 | J0 | F5 | F4 | F3 | F2 | F1 | F0 | R5 | S5 | R4 | S4 | R3 | S3 | R2 | S2 | R1 | S1 | R0 | S0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ↑ | ~ | 0 | ~ | 0 | ~ | 0 | ~ | 0 | ~ | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | ↑ | 1 | ~ | 0 | ~ | 0 | ~ | 0 | ~ | 0 | 0 | 1 | 0 | ~ |
| 2 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | ↑ | 1 | 1 | ~ | 0 | ~ | 0 | ~ | 0 | 0 | 1 | 0 | ~ | 0 | ~ |
| 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | ↑ | 1 | 1 | 1 | ~ | 0 | ~ | 0 | 0 | 1 | 0 | ~ | 0 | ~ | 0 | ~ |
| 4 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | ↑ | 1 | 1 | 1 | 1 | ~ | 0 | 0 | 1 | 0 | ~ | 0 | ~ | 0 | ~ | 0 | ~ |
| 5 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ↑ | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | ~ | 0 | ~ | 0 | ~ | 0 | ~ | 0 | ~ |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | ↓ | 0 | ~ | 0 | ~ | 0 | ~ | 0 | ~ | 0 | ~ | 1 | 0 |
| 7 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | ↓ | 0 | 0 | ~ | 0 | ~ | 0 | ~ | 0 | ~ | 1 | 0 | ~ | 0 |
| 8 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | ↓ | 0 | 0 | 0 | ~ | 0 | ~ | 0 | ~ | 1 | 0 | ~ | 0 | ~ | 0 |
| 9 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | ↓ | 0 | 0 | 0 | 0 | ~ | 0 | ~ | 1 | 0 | ~ | 0 | ~ | 0 | ~ | 0 |
| 10 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | ↓ | 0 | 0 | 0 | 0 | 0 | ~ | 1 | 0 | ~ | 0 | ~ | 0 | ~ | 0 | ~ | 0 |
| 11 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ↓ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | ~ | 0 | ~ | 0 | ~ | 0 | ~ | 0 | ~ | 0 |

Описание на Verilog HDL для отладки в ModelSim выполнено на основе оператора case.



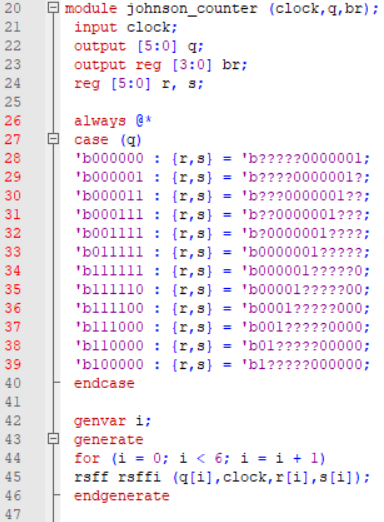


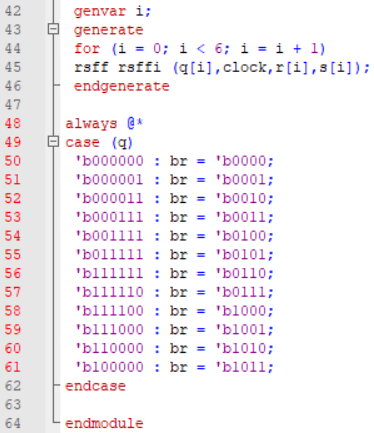
Испытательный стенд на Verilog HDL

Описание работы

RS-триггера

на Verilog HDL

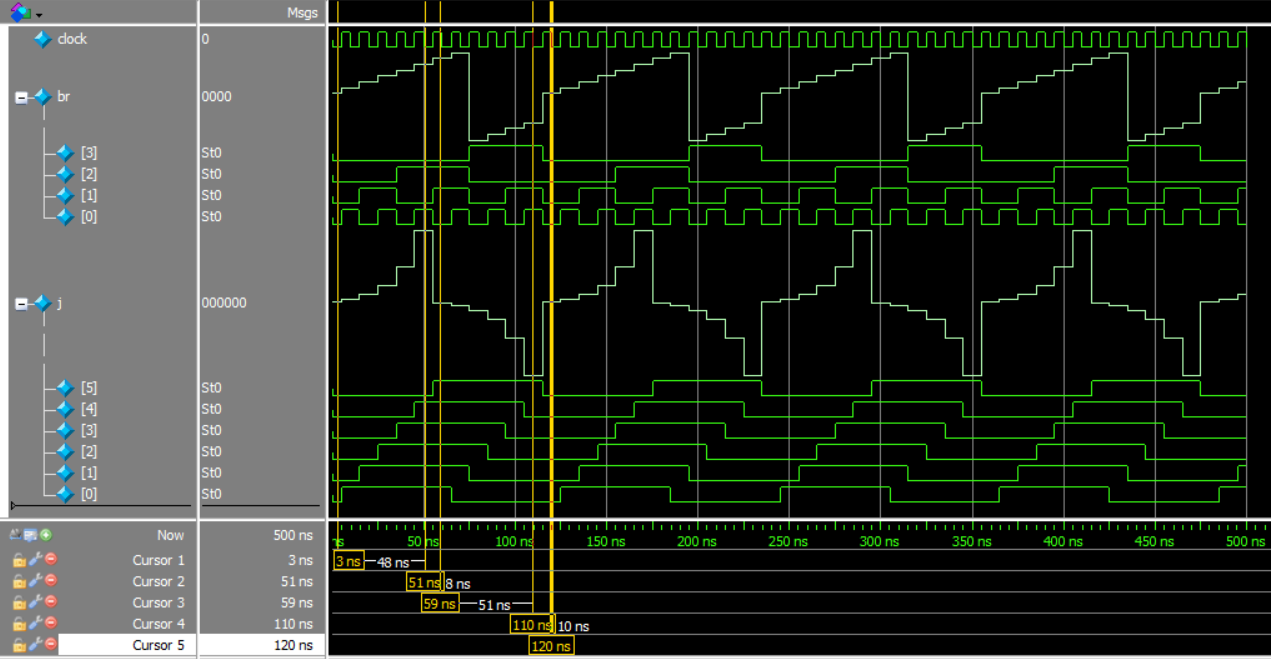




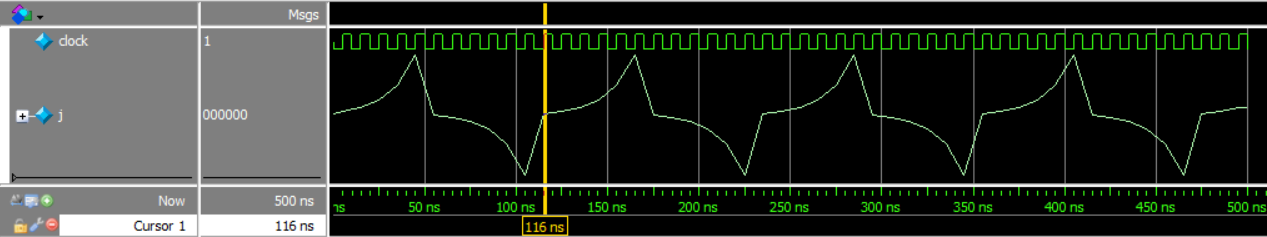
Описание на Verilog HDL. В данном модуле формируются информационные сигналы r и s и, затем передаются поразрядно в модуль rsff, в котором описана работа RS-триггера. Модуль rsff возвращает значения следующего состояния счетчика q также поразрядно.

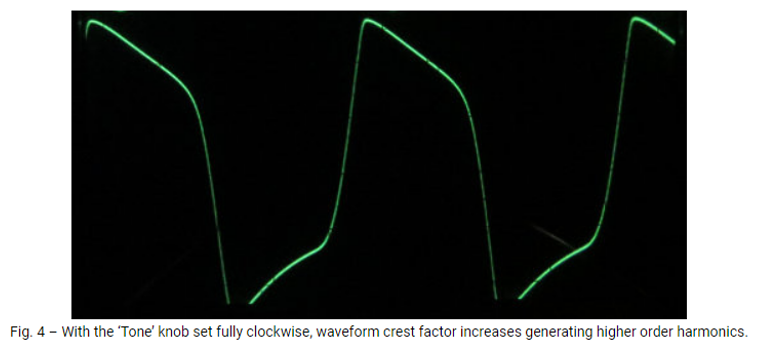
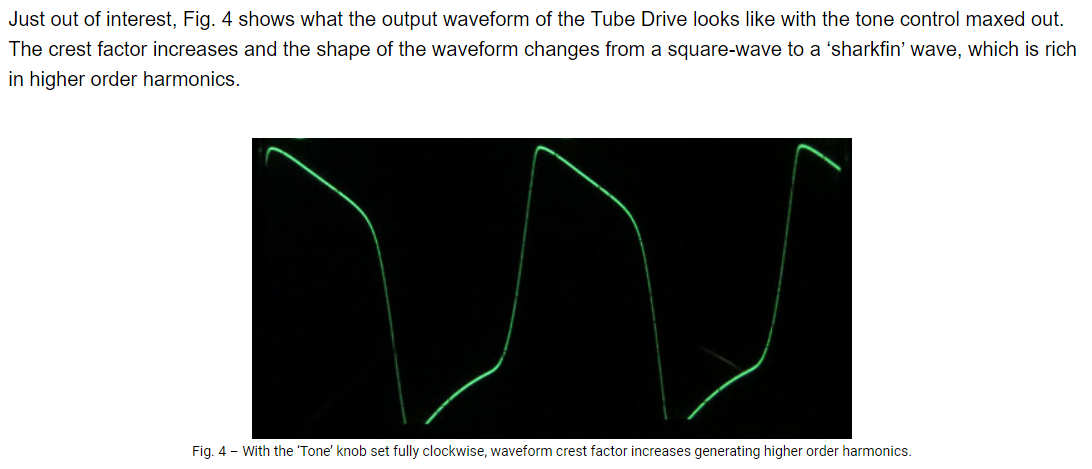
Описание на Verilog HDL. В данной части модуля формируются соответствующие каждому состоянию счетчика наборы двоично-арифметического кода. Двоично-арифметический код здесь имеет разрядность 4 (может принимать значения не более 15), так как соответствующий шестиразрядный код Джонсона имеет только 12 наборов.

На скриншоте представлены результаты моделирования в окне Wave. По неустановленной причине некоторые значения на данной осциллограмме отображаются ниже нуля. Чтобы показать, где находится нулевой уровень осциллограммы, курсор номер 5 установлен в положение, соответствующее нулевому значению br (ДАК) и j (код Джонсона). Тем не менее, в корректности работы программы можно убедиться, обратив внимание на сигналы br и j, отображенные поразрядно.

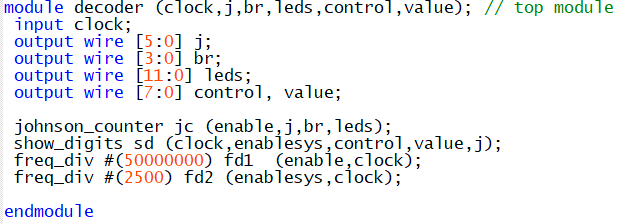
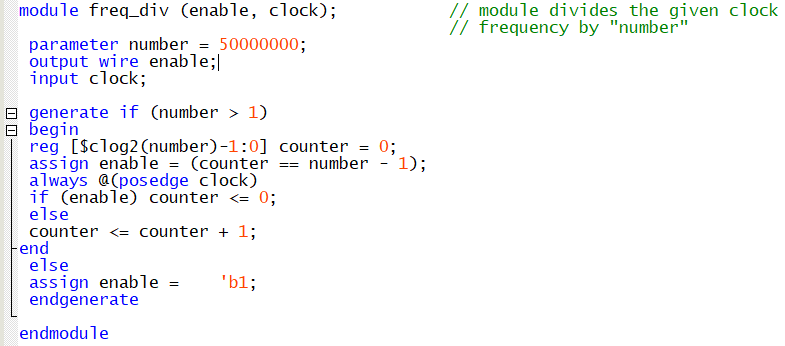


После интерполяции осциллограммы, сигнал стал отдаленно напоминать сигнал, снимаемый с выхода предусилителя для гитар типа Tube Drive. Реальные осциллограммы, снятые с этого устройства приведены на фотографиях ниже. На вход подается синусоидальная волна.



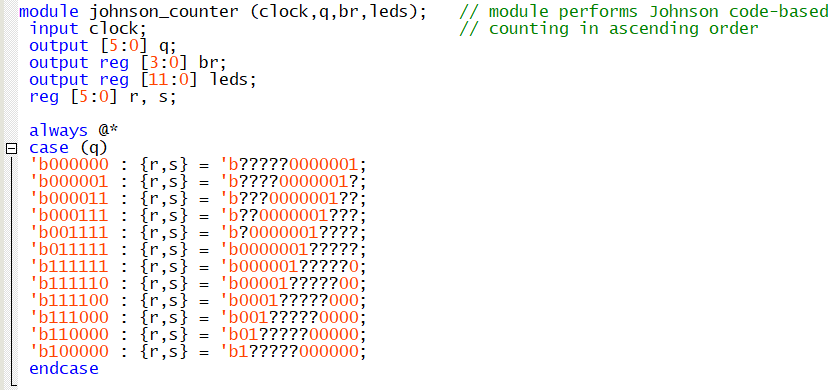
 

**Реализация шестиразрядного суммирующего счетчика в коде Джонсона на ПЛИС EP4CE10E22C8N**



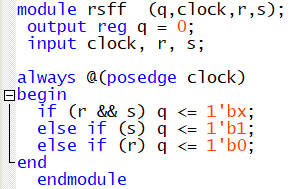
Описание на Verilog HDL модуля freq\_div. Модуль принимает в качестве входного сигнала тактовый сигнал частотой 50 МГц и возвращает импульсный сигнал частотой 50 / number МГц.

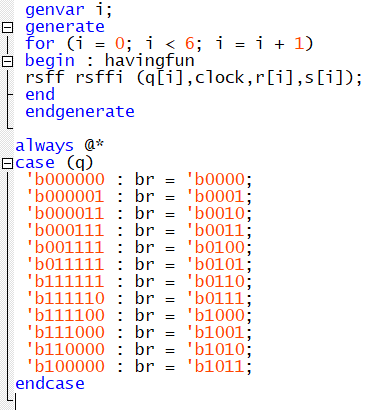
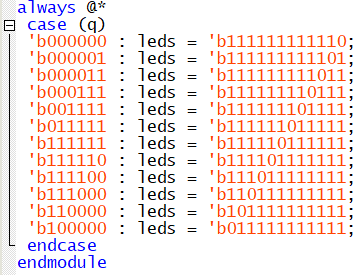
Описание на Verilog HDL основного модуля.



Описание RS-триггера на Verilog HDL.

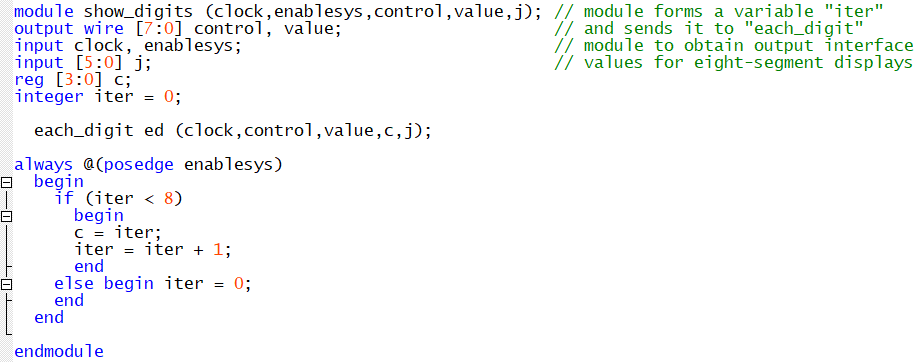
Описание на Verilog HDL модуля johnson\_counter. Модуль принимает в качестве входного сигнала импульсный сигнал пониженной частоты и возвращает текущий набор кода Джонсона q, соответствующее ему двоичное представление br и двенадцатиразрядный сигнал для светодиодов leds. Внутри оператора case формируются информационные сигналы r и s, которые затем будут переданы на RS-триггер.



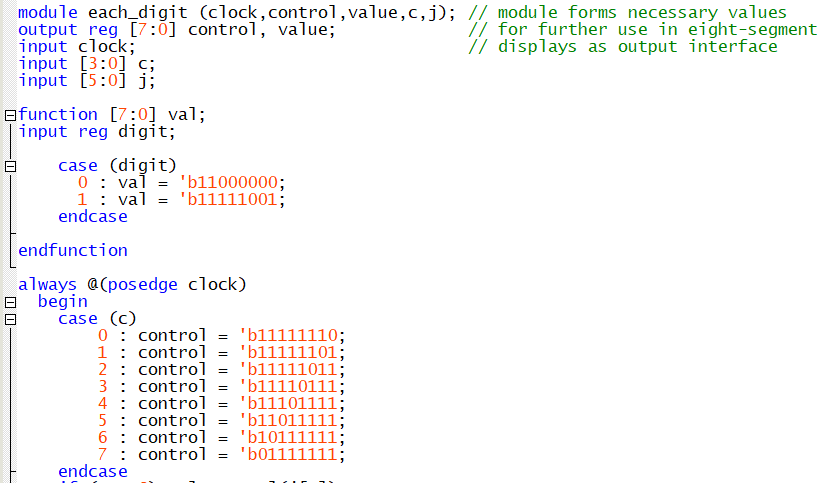
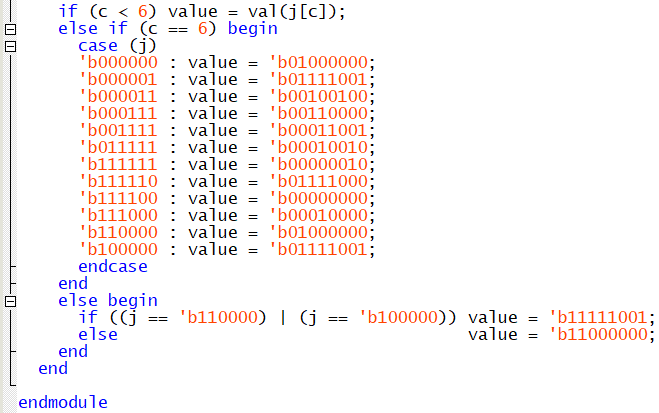


В данной части модуля с помощью оператора case формируются сигналы leds, которые впоследствии будут переданы на светодиоды.

Здесь информационные сигналы r и s передаются поразрядно в модуль rsff, в котором описана работа RS-триггера. Модуль rsff возвращает значения следующего состояния счетчика q также поразрядно. Также происходит формирование соответствующих наборов двоично-арифметического кода при помощи оператора case.



Модуль show\_digits принимает в качестве входных сигналов тактовый сигнал clock и импульсный сигнал пониженной частоты enablesys. Также на вход подается сигнал j, представляющий собой текущий набор кода Джонсона. Модуль возвращает два восьмиразрядных сигнала, первый из которых – control отвечает за активацию нужного разряда на блоке восьмисегментных индикаторов. Второй – value, содержит код, предназначенный для отображения соответствующей цифры на текущем разряде индикатора. Для получения control и value происходит обращение к модулю each\_digit. Ниже, при помощи оператора if, организован бесконечный цикл, который изменяет значение сигнала c от 0 до 7. Далее, c подается на вход модуля each\_digit.

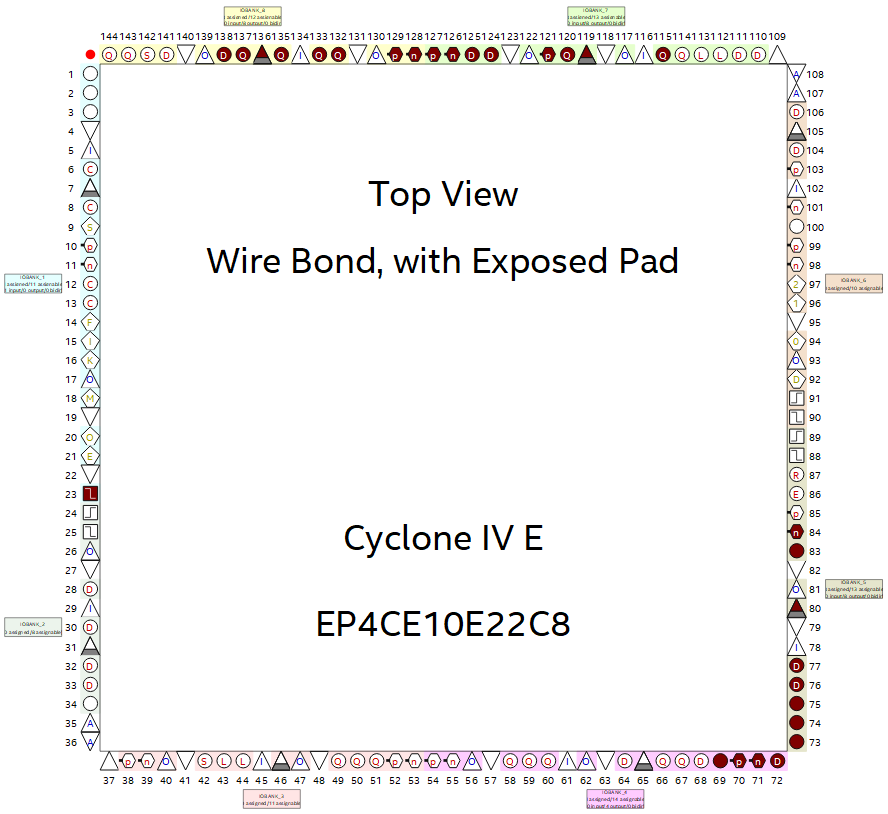
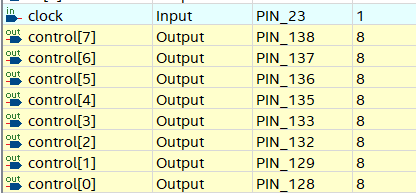
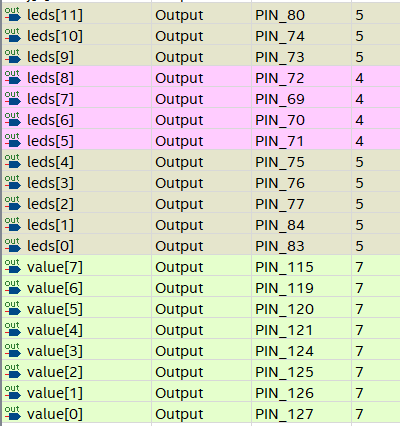


Модуль each\_digit принимает на вход импульсный сигнал пониженной частоты, значение текущего набора кода Джонсона j и c, сигнал содержащий значение от 0 до 7. Модуль возвращает два восьмиразрядных сигнала, первый из которых – control отвечает за активацию нужного разряда на блоке восьмисегментных индикаторов. При помощи оператора case, в зависимости от текущего значения c, соответствующего номеру разряда на блоке восьмисегментных индикаторов, формируется сигнал control. Также в этом участке кода описана функция val. Она принимает на вход значение одного разряда текущего набора кода Джонсона и возвращает восьмиразрядный сигнал, с помощью которого на соответствующем индикаторе будет отображена либо единица, либо ноль.

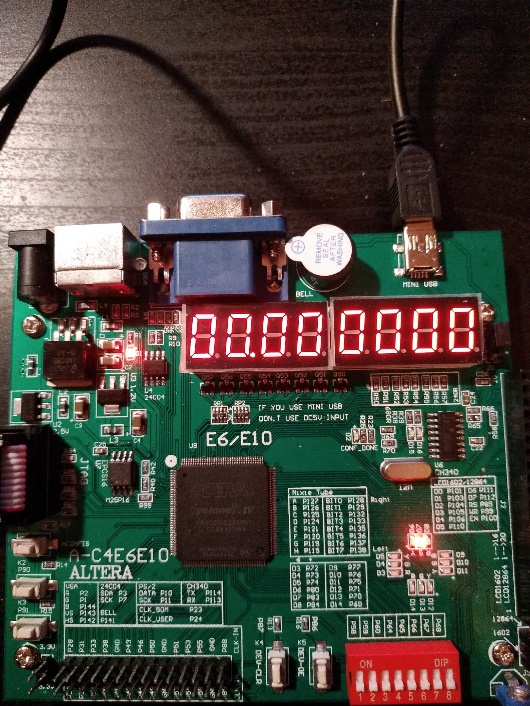
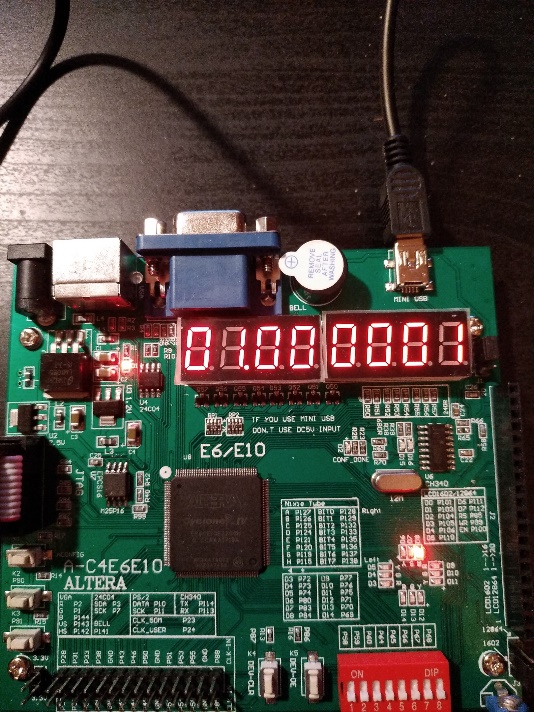
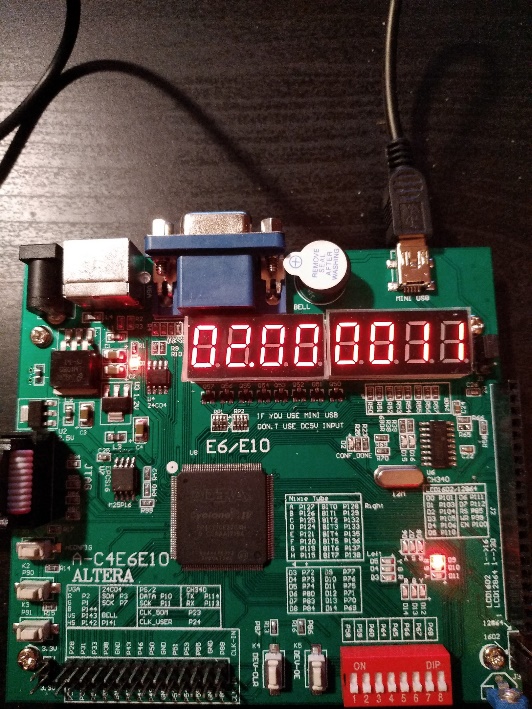
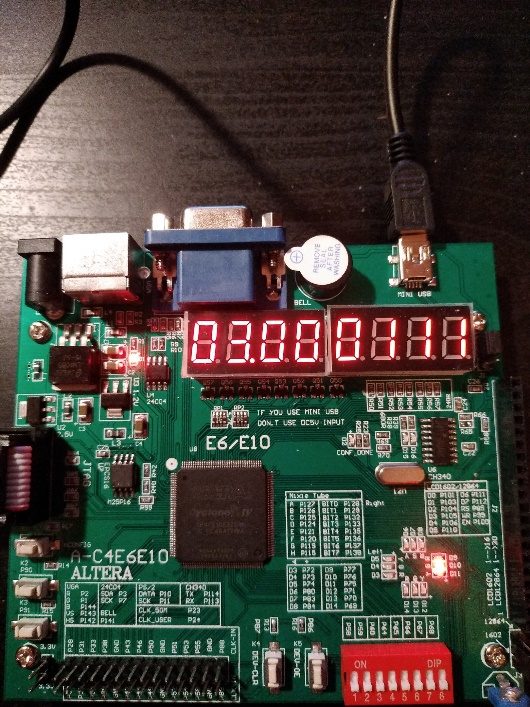
В данном участке кода происходит обращение к функции val, описанной выше. При условии, что номер разряда на блоке индикаторов не превышает 6, в функцию передается соответствующий разряд текущего набора кода Джонсона.

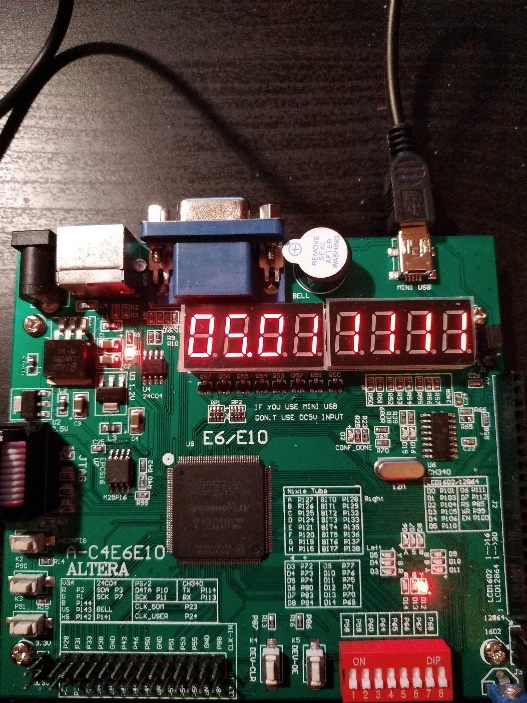
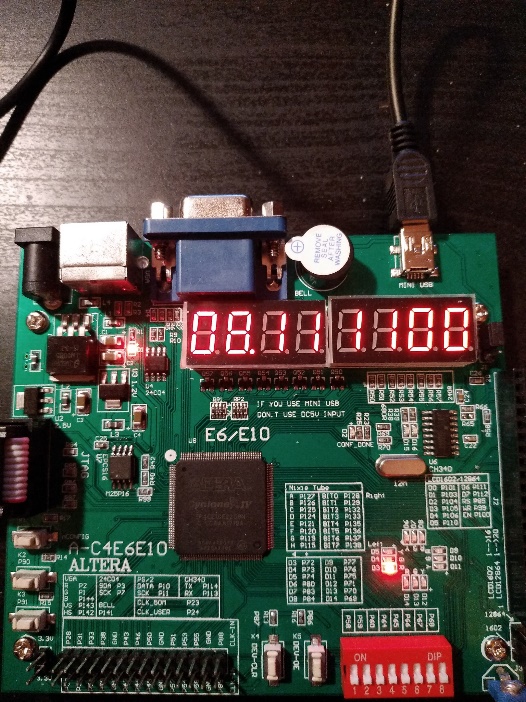
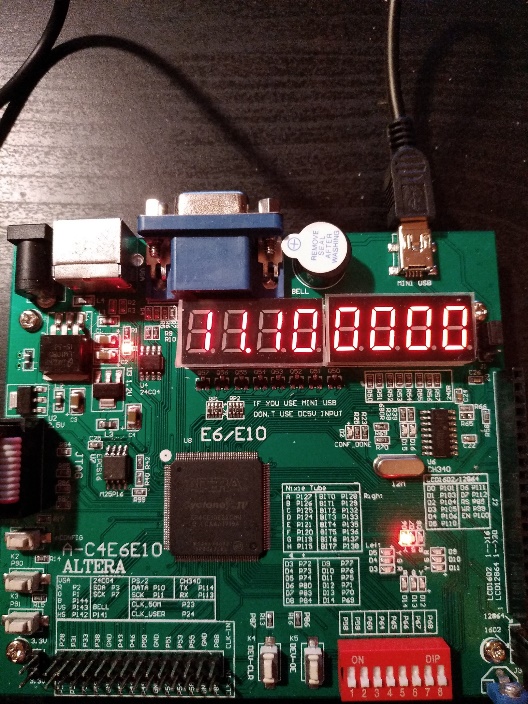
В оставшихся двух разрядах блока индикаторов отображается номер набора кода Джонсона в десятичном виде.

**Распиновка на ПЛИС Cyclone IV E EP4CE10E22C8 (скриншот из Quartus)**

На фотоснимках показана работа программы. На первых шести разрядах (с 0-го по 5-й) блока восьмисегментных индикаторов отображается текущий набор кода Джонсона. В разрядах 6 и 7 отображается соответствующий номер набора кода Джонсона в десятичной системе счисления. Светодиоды зажигаются поочередно по часовой стрелке, начиная с южного центрального светодиода (зажигается на нулевом наборе).

**Четырехразрядный вычитающий счетчик в коде Грея**

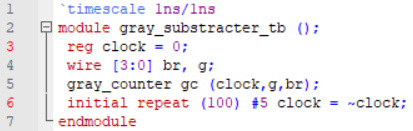
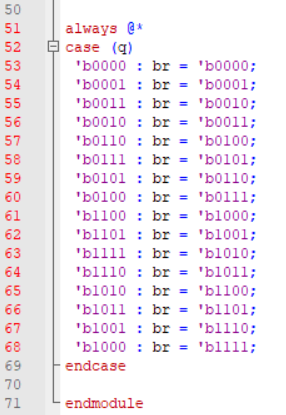
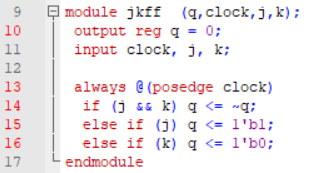
Для описания на Verilog HDL и отладки в ModelSim составим таблицу истинности. Для реализации счетчика в коде Грея был выбран JK-триггер. С целью получения необходимых для работы триггера информационных сигналов организуем таблицу истинности следующим образом:

* для удобства навигации в таблице запишем номера наборов в первый столбец;
* запишем текущее состояние и следующее состояние в следующие два больших столбца;
* для еще большего удобства работы с таблицей применим заливку различных оттенков таким образом, чтобы соответствующие разряды имели один и тот же цвет для всех больших столбцов;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Текущее  состояние | | | | Следующее состояние | | | | Функции  переходов | | | | Информационные  сигналы | | | | | | | |
| № | G3 | G2 | G1 | G0 | G3 | G2 | G1 | G0 | F3 | F2 | F1 | F0 | K3 | J3 | K2 | J2 | K1 | J1 | K0 | J0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | ↑ | 0 | 0 | 0 | ~ | 1 | ~ | 0 | ~ | 0 | ~ | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ↓ | ~ | 0 | ~ | 0 | ~ | 0 | 1 | ~ |
| 2 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | ↓ | 1 | ~ | 0 | ~ | 0 | 1 | ~ | 0 | ~ |
| 3 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | ↑ | ~ | 0 | ~ | 0 | 0 | ~ | ~ | 1 |
| 4 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | ↓ | 1 | 0 | ~ | 0 | 1 | ~ | 0 | ~ | ~ | 0 |
| 5 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | ↓ | ~ | 0 | 0 | ~ | ~ | 0 | 1 | ~ |
| 6 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | ↑ | 1 | ~ | 0 | ~ | 0 | ~ | 1 | 0 | ~ |
| 7 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | ↑ | ~ | 0 | 0 | ~ | ~ | 0 | ~ | 1 |
| 8 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | ↓ | 1 | 0 | 0 | 1 | ~ | 0 | ~ | ~ | 0 | ~ | 0 |
| 9 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | ↓ | 0 | ~ | 0 | ~ | ~ | 0 | 1 | ~ |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | ↓ | 1 | 0 | ~ | 0 | ~ | 1 | ~ | 0 | ~ |
| 11 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ↑ | 0 | ~ | 0 | ~ | 0 | ~ | ~ | 1 |
| 12 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | ↑ | 1 | 0 | 0 | ~ | ~ | 1 | 0 | ~ | ~ | 0 |
| 13 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | ↓ | 0 | ~ | ~ | 0 | 0 | ~ | 1 | ~ |
| 14 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | ↑ | 1 | 0 | ~ | ~ | 0 | ~ | 1 | 0 | ~ |
| 15 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | ↑ | 0 | ~ | ~ | 0 | ~ | 0 | ~ | 1 |

* получим функции переходов для каждого перехода из текущего состояния в следующее;
* при помощи словаря переходов отыщем информационные сигналы, соответствующие полученным функциям переходов.

Описание на Verilog HDL для отладки в ModelSim выполнено на основе оператора case.

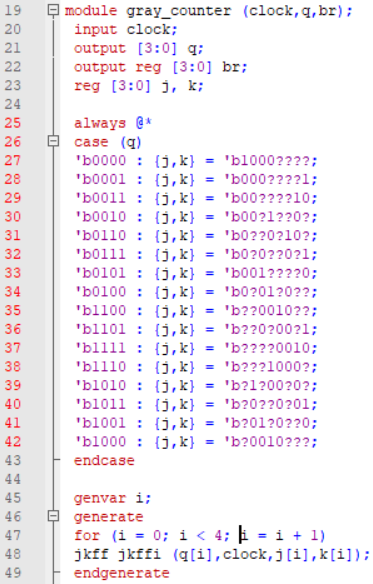


Испытательный стенд на Verilog HDL

Описание работы

JK-триггера

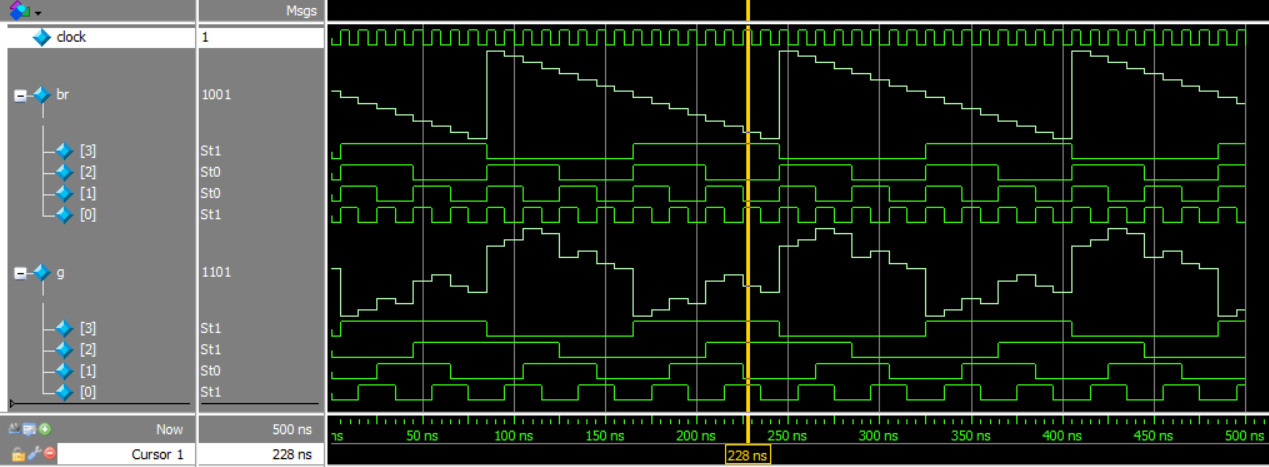
на Verilog HDL



Описание на Verilog HDL. В данном модуле формируются информационные сигналы j и k и, затем передаются поразрядно в модуль jkff, в котором описана работа JK-триггера. Модуль jkff возвращает значения следующего состояния счетчика q также поразрядно.

Описание на Verilog HDL. В данной части модуля формируются соответствующие каждому состоянию счетчика наборы двоично-арифметического кода.

На скриншоте представлены результаты моделирования в окне Wave.



**Вывод**

На Verilog HDL были описаны суммирующий счетчик в коде Джонсона и вычитающий счетчик в коде Грея.

Для индикации текущих наборов в реверсивном счетчике, были использованы следующие новые (для автора работы) методы:

1. Применение делителя тактовой частоты. Так как тактовая частота составляет на ПЛИС EP4CE10E22C8N 50 МГц, при индикации значений на такой частоте, визуальное восприятие результата не представляется возможным. По этой причине был использован модуль freq\_div, получающий на основе тактовой частоты импульсный сигнал пониженной частоты.

2. Индикация наборов кода, состоящих из нескольких разрядов на блоке восьмисегментных индикаторов. Для достижения работы индикации было решено воспользоваться делителем частоты повторно, создав дополнительный экземпляр модуля freq\_div, передав в него в качестве параметра число, меньшее, чем коэффициент делителя, использованный для смены наборов в коде Джонсона и в коде Грея.

**Литература**

1. [ru.wikipedia.org › wiki › Код\_Грея](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4_%D0%93%D1%80%D0%B5%D1%8F" \l ":~:text=%D0%9A%D0%BE%D0%B4%20%D0%93%D1%80%D0%B5%CC%81%D1%8F%20%E2%80%94%20%D0%B4%D0%B2%D0%BE%D0%B8%D1%87%D0%BD%D1%8B%D0%B9%20%D0%BA%D0%BE%D0%B4%2C%20%D0%B8%D0%BD%D0%B0%D1%87%D0%B5,%D1%86%D0%B8%D1%84%D1%80%D0%BE%D0%B9%20%D0%B2%20%D0%BE%D0%B4%D0%BD%D0%BE%D0%BC%20%D0%B4%D0%B2%D0%BE%D0%B8%D1%87%D0%BD%D0%BE%D0%BC%20%D1%80%D0%B0%D0%B7%D1%80%D1%8F%D0%B4%D0%B5.)

2. [Иосиф Каршенбойм. «Краткий курс HDL»](http://iosifk.narod.ru/hdl_coding/verilog.htm)

3. <http://www.asic-world.com/>

4. [Chris Spear. «SystemVerilog for Verification»](http://www.amazon.com/SystemVerilog-Verification-Learning-Testbench-Language/dp/144194561X)

5. [Altera. «Quartus Handbook»](https://www.altera.com/en_US/pdfs/literature/hb/qts/qts-qps-handbook.pdf)