Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

“Санкт-Петербургский государственный политехнический университет”

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

**Отчёт по лабораторной работе №7**

по дисциплине “Схемотехника операционных устройств”

на тему “Исследование регистров”

Выполнил студент группы 5130901/20003

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Вагнер А.А.

Принял преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Киселёв И.О.

“\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 года

Санкт-Петербург

2024

Цель работы

− исследование режимов загрузки и сдвига данных в регистрах;

− исследование регистров с обратными связями;

− исследование типовых функциональных устройств с использованием регистров;

− получение навыков по тестированию и отладке устройств с регистрами.

Вариант задания:

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Ход работы

**1. Исследование работы сдвигающего регистра на основе мегафункции LPM\_SHIFTREG**

В соответствии с вариантом была создана мегафункция LPM\_SHIFTREG с разрядностью 16.

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рис. 1 – Схема двигающего регистра

Убедимся в корректности работы данного регистра при помощи моделирования. С помощью входов data[15..0] и load в регистр загружаются данные. Shiftin определяет, будут ли освобождающиеся значения заполняться нулями или единицами. Выход shiftout показывает, какой символ был смещён на данном такте.

Изображение выглядит как текст, число, Параллельный, линия

Автоматически созданное описание

Рис. 2 – Моделирование работы сдвигающего регистра на основе мегафункции LPM\_SHIFTREG

2. Разработка на базе мегафункции LPM\_SHIFTREG регистра, позволяющего загружать информацию и сдвигать её в обоих направлениях

На базе мегафункции LPM\_SHIFTREG построим схему регистра, позволяющего сдвигать данные в нём в оба направления с помощью входа control[1..0]. Режимы работы:

* control[1] = control[0] = 0 – загрузка информации в регистр
* control[1] = 0; control[0] = 1 – сдвиг вправо
* control[1] = 1; control[0] = 0 – сдвиг влево

Работа регистра заключается в том, что при сдвиге влево он работает как обычный регистр. При сдвиге направо, он загружает в себя же данный, которые хранятся внутри регистра на данный момент времени, без младшего бита и shift\_in\_r в страшем бите. Таким образом, на каждом такте происходит подмена с q[15..0] на shift\_in\_r и q[15..1].

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Рис.3 – Схема регистра, сдвигающего в оба направления.

Изображение выглядит как линия, число, текст, Параллельный

Автоматически созданное описание

Рис.4 – Моделирование работы реверсного регистра, ч.1.

Изображение выглядит как текст, линия, График, снимок экрана

Автоматически созданное описание

Рис.5 – Моделирование работы реверсного регистра, ч.2.

3. Исследование счётчика Джонсона

Построим на основе мегафункции LPM\_SHIFTREG счётчик Джонсона. Особенность счётчика Джонсона заключается в том, что каждое соседнее значение счётчика в любой момент времени отличается от текущего всего на один бит. Это достигается путём сдвига текущего значения в счётчике влево, добавляя в младшие разряды инвертированные значения shift\_out, как видно при моделировании работы данной схемы.

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Рис.6 – КС счётчика Джонсона

Изображение выглядит как текст, линия, снимок экрана, число

Автоматически созданное описание

Рис.7 – Моделирование счётчика Джонсона

4-5. Генератор псевдослучайных чисел.

Просмтроим на основе мегафункции LPM\_SHIFTREG генератор псевдослучайных двоичных числе.

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание

Рис.8 – КС генератора псевдослучайных чисел

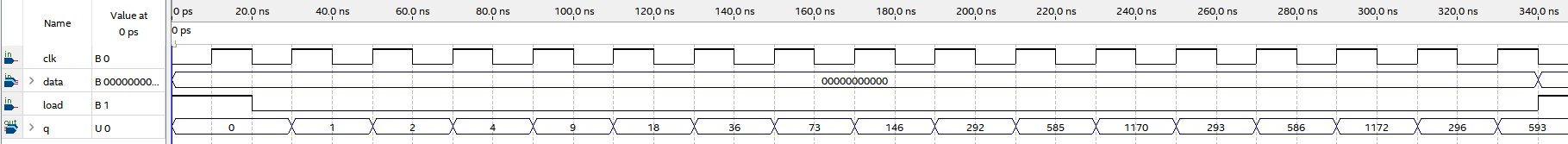


Рис.9 – моделирование генератора псевдослучайных чисел ч.1

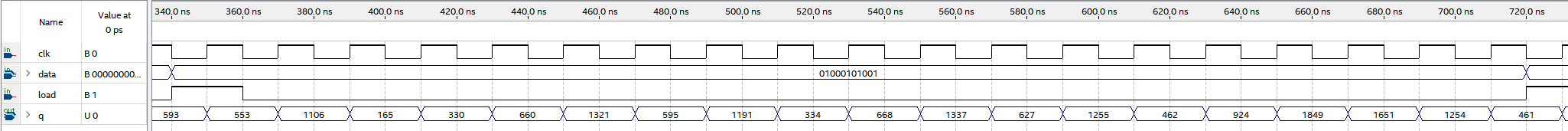


Рис.10 – моделирование генератора псевдослучайных чисел ч.2

6. Определение периода формирования псевдослучайных чисел.

Подсчитаем количество повторяющихся сигналов на псевдослучайном генераторе при входном значении 1, а также произведём моделирование работы устройства. К счётчику был добавлен вход aclr для того, чтобы подсчёт количества тактовых сигналов начинался после того, как сигнал load станет равным 0, и начнётся генерация чисел.

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Рис.11 – КС для счёта периода

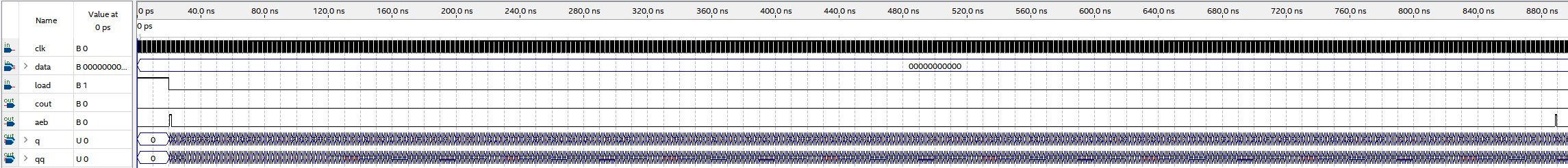


Рис.12 – Моделирование КС для счёта периода

Изображение выглядит как линия, текст, число, Шрифт

Автоматически созданное описание

Рис.13 – Момент начала цикла на модели

Длина цикла для моего генератора – 869 итераций, что соответствует условию вхождения в интервал

7. Исследование устройства, преобразующего последовательный код в параллельный.

Создадим схему преобразователя последовательного кода в параллельный. Началом слова будет считаться последовательно семи единиц «111111». Код последовательно заносится в регистр Shift\_RG. После каждого добавления нового элемента последовательного кода, верхняя часть схемы проверяет, заполнены ли младшие шесть битов данного регистра единицами. Если условие выполняется, то D-триггер переводится в состояние «1», и начинается подсчёт 17-го такта, по окончанию которого подаётся сигнал cout = 1 (strobe = 1), данные из Shift\_RG загружаются в Out\_RG, а счётчик и Dтриггер обнуляются.

Изображение выглядит как текст, диаграмма, снимок экрана, число

Автоматически созданное описание

Рис.14 – КС перевода последовательного кода в параллельный

Изображение выглядит как текст, линия, число, График

Автоматически созданное описание

Рис.15 – Моделирование КС ч.1

В данном случае на вход DS подаётся сигнал 1111111011111111111000. Первые 6 битов переводят D-триггер в состояние 1. Следовательно ожидаемое состояние Q – 1011111111111000.

Изображение выглядит как текст, снимок экрана, линия, число

Автоматически созданное описание

Рис.16 – Моделирование КС ч.2

8. Разработка фрагмента передатчика для универсального асинхронного канала приёмопередачи.

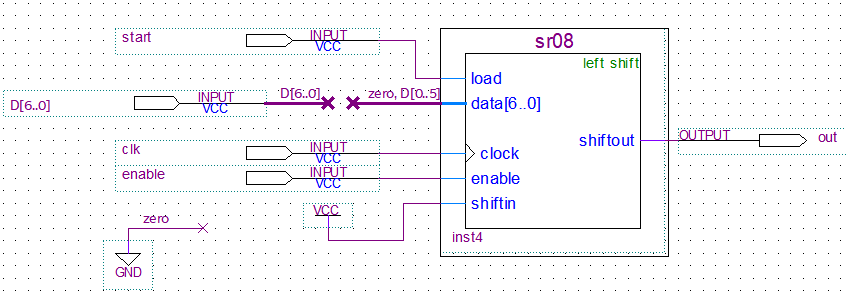


Рис.17 – КС фрагмента передатчика

Изображение выглядит как текст, снимок экрана, число, линия

Автоматически созданное описание

Рис.18 – Моделирование КС фрагмента передатчика

Очевидно, что код регистра возвращает “101001”, то есть сигнал, переданный на D, транслируемый последовательно младшими битами вперёд

Вывод

В ходе данной работы были исследованы режимы загрузки и сдвига данных в регистрах, а также устройства основанные на работе регистров, такие как счётчик Джонсона, генератор псевдослучайных чисел, преобразователь последовательного кода в параллельный и фрагмент универсального асинхронного канала приёмопередачи.