

## Регистры

При обработке сигналов в цифровых устройствах часто необходимо оперировать не отдельными битами, а наборами (векторами) битов, или, другими словами, – не одно-, а многоразрядными числами. Для этого триггеры объединяются в группы, которые называются регистрами. Регистры широко применяются как для собственно хранения данных, так и для реализации отдельных устройств обработки информации.

### Виды и принципы работы регистров

Регистр – это последовательностное цифровое устройство, предназначенное для хранения многоразрядного двоичного числа. Обычно регистры имеют регулярную структуру и состоят из однотипных элементов. Каждый из таких элементов (разряд регистра) предназначен для хранения одного разряда двоичного числа. Разряд регистра представляет из себя простейший элемент памяти, выполненный на триггере того или иного типа. Регистры широко используются при реализации алгоритмов формирования и обработки сигналов в системах передачи данных.

Регистр может работать в режимах записи, хранения и считывания чисел. В режиме записи в регистр вводится двоичное число. В режиме хранения записанное число остается без изменения, а при считывании – число передается из регистра в другие устройства.

По способам записи и считывания регистры делятся на последовательные (регистры сдвига), параллельные, последовательно-параллельные и параллельно-последовательные.

По типу обрабатываемых данных регистры бывают одноразрядными и многоразрядными. В первых обрабатываемой единицей данных является бит, а у вторых – целое многоразрядное слово.

Операции записи/чтения могут выполняться как по фронту/срезу тактовой частоты, так и при наличии дополнительного разрешающего сигнала. На рис. 1–4 изображены синхронные регистры, запись/чтение которых выполняется по тактовой частоте (отсутствует дополнительный сигнал разрешения).

В последовательных регистрах (регистрах сдвига, сдвигающих регистрах) запись и считывание сохраненных в регистре данных выполняется по одному разряду за один раз (обычно, за такт). Т.е. при наличии разрешающего сигнала на входе регистра выполняется выгрузка одного (старого) элемента и загрузка в регистр одного (нового). Регистры сдвига применяются при реализации алгоритмов помехоустойчивого кодирования, блоков генерирования псевдослучайных последовательностей для расширения спектра сигналов (UMTS), скремблирования или временной синхронизации (UMTS, WiFi, LTE и др.).

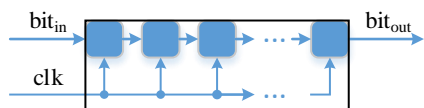


Рисунок 1. Одноразрядный синхронный регистр сдвига. Сдвиг выполняется по фронту тактовой частоты.

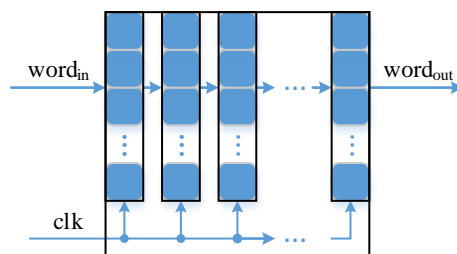


Рисунок 2. Многоразрядный синхронный регистр сдвига. Сдвиг выполняется над целыми словами.

В параллельных регистрах по разрешающему сигналу выполняется запись и считывание сразу всех ячеек регистра. Параллельные регистры часто применяются для реализации линий задержки многоразрядных данных.

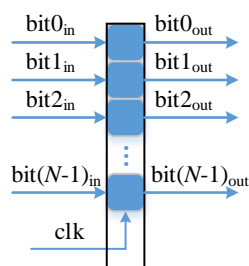


Рисунок 3. Параллельный одnorазрядный регистр.

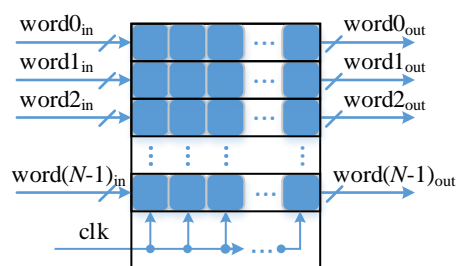


Рисунок 4. Параллельный многоразрядный регистр.

Параллельно-последовательные и последовательно-параллельные регистры являются гибридными по способу записи и чтения. Такие регистры обычно применяются для преобразования параллельного потока данных (многоканального) в последовательный и наоборот.

### Пример 1. Применение регистров в помехоустойчивом кодировании

Помехоустойчивое кодирование применяется для обнаружения и/или исправления ошибок в сообщении, возникающих при передаче данных по каналам с шумами, помехами или замираниями. В современных беспроводных цифровых системах передачи данных/связи то или иное помехоустойчивое кодирование применяется почти<sup>1</sup> всегда. Существует большое количество разнообразных кодов, обнаруживающих и исправляющих ошибки, которые возникают в сообщениях при их передаче по каналам связи. Важным классом таких кодов являются сверточные коды: принцип их работы построен на основе свертки (со сложением по модулю два) кодируемой полезной двоичной последовательности с определенной импульсной характеристикой устройства на основе регистра сдвига.

Например, кодер<sup>2</sup> одного из наиболее популярных на сегодняшний день сверточных кодов (171,133) строится на основе простейшего регистра сдвига с отводами на два сумматора по модулю два (см. рис. 5). Числа в описании кода – (171,133) – описывают, из каких ячеек сдвигающего регистра берутся отводы на сумматоры по модулю два. Числа 171 и 133 записаны в восьмеричной системе счисления, их количество (два) говорит о том, что сумматоров в кодере два (т.е. на каждый входной информационный символ будет генерироваться два символа канального алфавита на выходе кодера). Если перевести каждую цифру восьмеричных чисел 171 и 133 в двоичную систему счисления<sup>3</sup> (MSB слева), то от разрядов, в которых присутствует единица, берется отвод на соответствующий сумматор.

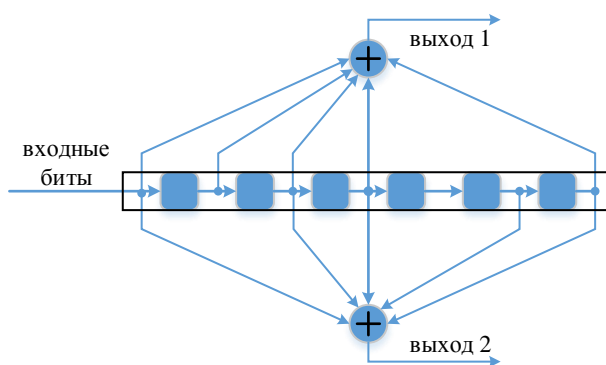


Рисунок 5. Структура кодера сверточного кода (171,133).

Описанный сверточный код (171,133) почти всегда используется в беспроводных цифровых системах передачи данных как вариант по умолчанию. Отметим также, что сверточные коды

<sup>1</sup> “Почти” относится к режимам работы системы, а не к системе целиком.

<sup>2</sup> Вообще, кодеры всех сверточных кодов строятся на основе регистров сдвига с отводами на сумматоры по модулю два.

<sup>3</sup> Результатом этой операции будут следующие последовательности битов: 171 => 001 111 001, 133 => 001 011 011. Старшие нулевые два бита последовательностей не используются.

применяются в качестве составных элементов более сложных сверточных турбокодов, обладающих замечательными корректирующими способностями. Сверточные турбокоды применяются в таких системах связи как WiMAX, UMTS, LTE, а также в обратных каналах DVB RCS-1, 2 спутниковых систем цифрового вещания.

## Пример 2. Применение регистров в генераторах псевдослучайных последовательностей и скремблерах

Генераторы **псевдослучайной последовательности** (ПСП) генерируют последовательности, имеющие определенные статистические свойства (постулаты Голомба). При рассмотрении “хорошей” ПСП кажется, что ее генерировал случайный источник. Далее сгенерированная ПСП может применяться для решения различных задач: скремблирования, расширения спектра, обнаружения и синхронизации (частотной и временной).

**Скремблер** (рандомайзер) придает передаваемой последовательности символов псевдослучайный характер: в процессе скремблирования передаваемая двоичная последовательность суммируется (в аддитивных скремблерах, есть еще мультипликативные) по модулю два со сгенерированной ПСП. В зависимости от системы передачи данных, скремблер может применяться для решения задач тактовой синхронизации, улучшения формы спектра, снижения вероятности появления кодовых слов с низким весом на входе кодеров.

Структура аддитивного скремблера системы WiMAX представлена на рис. 6. Видно, что скремблер состоит из двух частей: генератор ПСП (регистр сдвига и линии обратной связи с сумматорами по модулю два) и сумматор генерируемой ПСП с входной последовательностью.

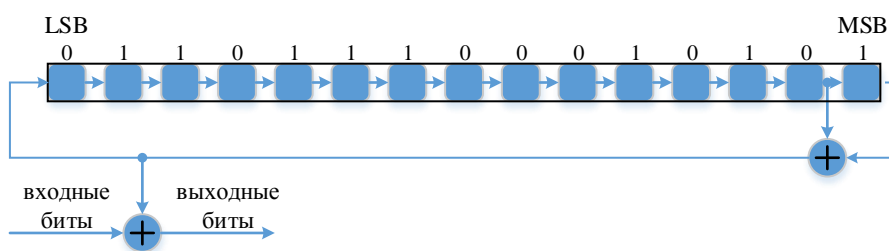


Рисунок 6. Аддитивный скремблер системы WiMAX.

При **расширении спектра** каждому передаваемому биту ставится в соответствие большое (сотни, иногда тысячи) количество символов ПСП<sup>4</sup>. За счет этого спектр сигнала “размазывается” по большой полосе частот, а сам сигнал становится шумоподобным и трудно обнаруживаемым. Расширение спектра используется в военных системах связи для противодействия преднамеренным помехам, а в гражданских системах связи – для борьбы с промышленными и бытовыми помехами (WiFi – 802.11b), а также для реализации кодового<sup>5</sup> разделения абонентов (UMTS).

## Пример 3. Линии задержки

При реализации алгоритмов обработки сигналов в FPGA часто необходимо задержать данные на один или несколько тактов. Это необходимо, когда обработка данных осуществляется одновременно по нескольким ветвям с разной задержкой, после чего результаты из каждой ветви обрабатываются совместно. Также бывает необходимо задерживать и одноразрядные данные: сигналы разрешения, запуска, data\_valid и т.п.

<sup>4</sup> При описанном методе расширения спектра сигнала обычно используется подвид ПСП, который называется *M*-последовательностями. *M*-последовательности обладают очень хорошими автокорреляционными свойствами (низкий уровень боковых пиков автокорреляционной функции).

<sup>5</sup> Напомним, что существует три основных метода уплотнения абонентов: временное, частотное и кодовое.

На рис. 7 показана линия задержки восьмиразрядных двоичных чисел на 5 тактов. На каждом такте осуществляется выгрузка из регистра старого числа и загрузка нового.

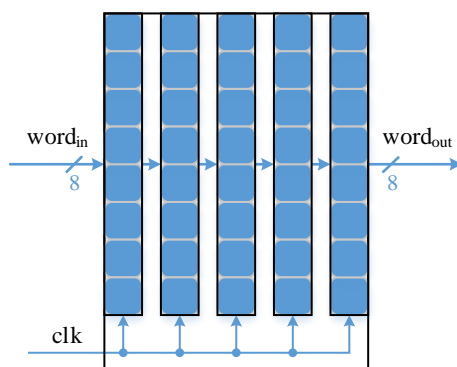


Рисунок 7. Линия задержки восьмиразрядных двоичных чисел на 5 тактов.

Задержку сигналов на один такт (синхронный регистр сдвига из одного элемента) также удобно применять, когда из длинного сигнала разрешения необходимо сформировать короткий, длительностью в один такт, строб. Схема формирования такого строба и поясняющие временные диаграммы представлены на рис. 8.

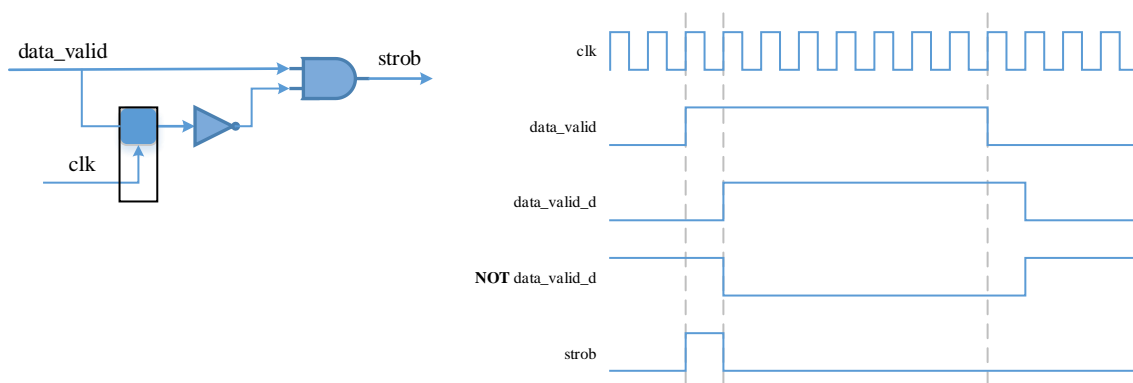


Рисунок 8. Формирование строба.

Обратим внимание на то, что для реализации линии задержки может использоваться как логика общего назначения, так и встроенные блоки памяти FPGA. Обычно короткие линии задержки реализуются на логике общего назначения, а длинные – на блоках памяти.

## Особенности FPGA

Основным ресурсом FPGA Xilinx 7-о поколения является конфигурируемый логический блок (Configurable Logical Block, CLB). В каждом логическом блоке содержится два модуля, которые называются slice, не связанных друг с другом. Логический блок подключен к матрице подключений (рис. 9<sup>6</sup>), которая в свою очередь подключена к глобальной матрице соединений.

<sup>6</sup> [www.xilinx.com](http://www.xilinx.com)

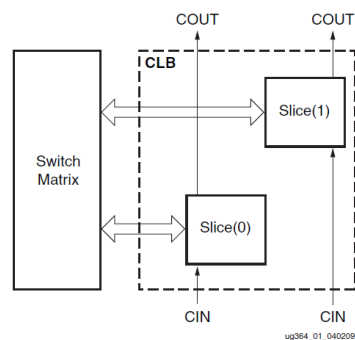


Рис. 9. Структура конфигурируемого логического блока

Логические блоки (и slice'ы) в ПЛИСе объединяются в столбцы, при этом slice, которые находятся в одном столбце соединены очень быстрыми линиями для реализации цепей переноса (необходимы, например, при выполнении арифметических операций).

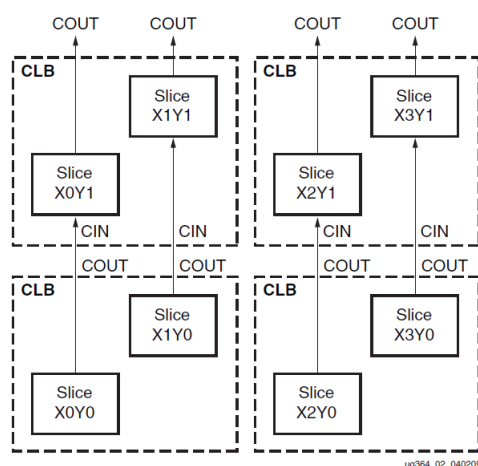


Рис. 10. Объединение логических блоков в столбцы

Для возможности реализации логических и арифметических функций, а также памяти в состав slice входят:

- 4 таблицы истинности (генераторы функций);
- 8 элементов хранения (триггеры);
- мультиплексоры;
- линии переноса.

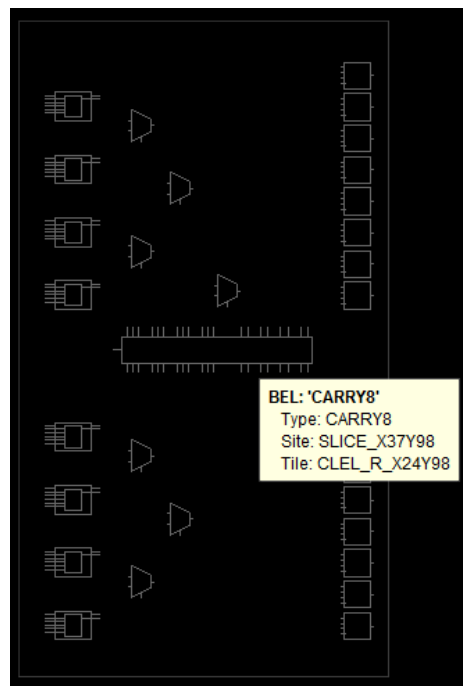
Существует два типа slice: SLICEM и SLICEL. SLICEM включает в себя поддержку двух дополнительных функций: хранение данных в распределенной памяти и сдвиг данных в 32-х разрядном сдвигающем регистре. В каждом конфигурируемом логическом блоке может быть либо два slice типа SLICEL, либо один slice типа SLICEL и один slice типа SLICEM. Каждый второй столбец CLB содержит SLICEM. Всего в ПЛИС slice типа SLICEM меньше, чем slice типа SLICEL.

Генераторы функций реализованы в виде 6-входовых таблиц истинности. Каждая таблица истинности имеет 6 независимых входов и 2 независимых выхода. Каждая таблица истинности может реализовывать 2 логических функции, если входы для них общие. Вывод данных осуществляется непосредственно с выходов slice либо через соответствующие мультиплексоры. В slice типа SLICEM в таблицах истинности могут быть также реализованы 256 бит распределенной RAM или 128-битный сдвигающий регистр.

На рис. 11 представлен внешний вид slice Xilinx FPGA серии Ultrascale. Фактически это удвоенные slice предыдущей серии. При увеличенном масштабе в левой части slice видны 8 таблиц истинности, в правой – 16 триггеров, а посередине – мультиплексоры выходов таблиц истинности и цепь переноса.



а) Slice внутри Xilinx FPGA серии Ultrascale



б) Slice. Увеличенный масштаб

Рис. 11. Slice внутри Xilinx FPGA серии Ultrascale