

Очень часто читатели моего сайта задают один и тот же вопрос: "Как в DAC 1x соединить ресивер и ЦАПы, разделяя общий цифровой поток на правую и левую части?". При этом чаще всего добавляя, что в цифровой технике не сильны :))).

Чаще всего я отвечаю, что это просто, нужно лишь просто напросто **_ВНИМАТЕЛЬНО_** прочитать даташиты от приемника цифрового сигнала и м\с ЦАП и совсем немножко подумать, а главное - разобраться как это все работает.

Существуют два способа это проделать.

Способ первый: "прерывание тактовой".

Приемник CS8412(14) и пара цап PCM56. У 8412(14) имеем три "ноги" SCK, FSYNC, SDATA. У PCM56 три "ноги": BCK, LE, DATA.

Начинаем с м\с ЦАП - данные отсчета (DATA) загружаются во внутренний регистр ЦАП по тактам BCK, после того как данные будут загружены в этот регистр, перепад уровня на "ноге" управления обновлением ЦАП (LE) заставит ЦАП выдать ток на аналоговом выходе, соответствующий цифровому отсчету и удерживать его до следующего обновления.

Таким образом делаются два важных вывода:

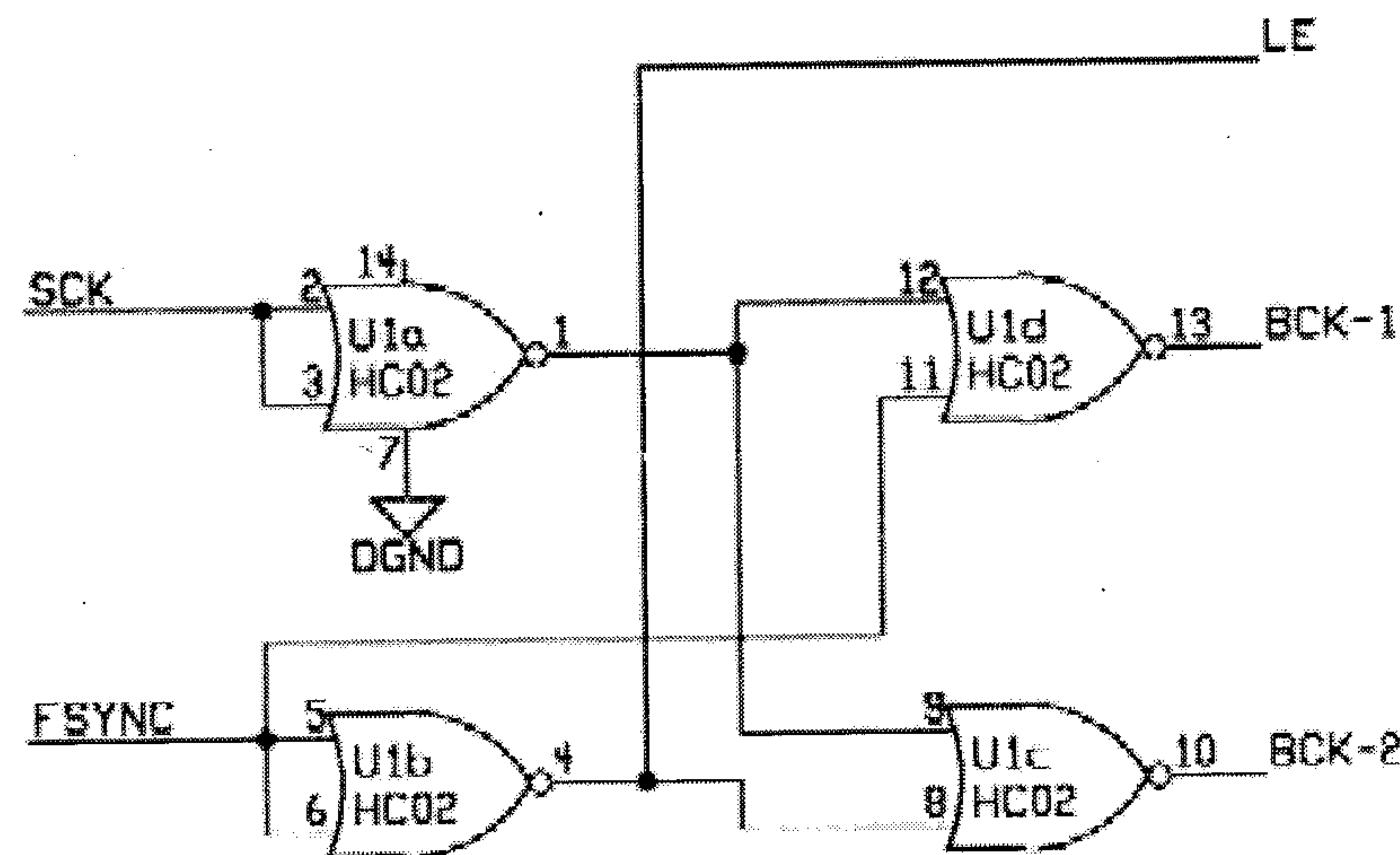
- данные грузятся в ЦАП по сигналам BCK и при их отсутствии всякое "шевеление" данных на ноге DATA ни к чему не приводит, а также что ножки DATA и LE ЦАП левого и правого канала соединяются вместе.
- при наличии сигнала BCK в регистр попадут только последние N-бит данных, соответствующих емкости регистра и разрядности ЦАП.

Исходя из вышеописанного следует, что формат выходных данных ресивера должен быть таким, чтобы данные отсчетов правого и левого канала были смещены к правой части сигнала FSYNC. Т.е. Right-Justified Format.

Теперь смотрим как идут сигналы в этом формате от ресивера: при высоком уровне сигнала FSYNC идут "левые" отсчеты, при низком правые. Значит всего-то и нужно поставить два логических элемента, пропускающих тактовую SCK от 8412(14) на соответствующие ноги BCK ЦАП правого и левого каналов в

зависимости от уровня FSYNC. Сам сигнал FSYNC будет служить сигналом обновления м\с ЦАП - LE, только его нужно проинвертировать, т.к. обновление ЦАП PCM56 наступает при перепаде от высокого уровня к низкому. И помня о последних 16 -битах загружающихся в регистр, можно сэкономить элемент, отключающий тактовую для ЦАП правого канала - данные и так будут последними.

Для ничего не понявших привожу "конкретную" схему на 74HC02:



В следующий раз я расскажу о регистрах сдвига и втором способе :)))

Sp10G

Как сделать самый "интересный" узел 1x DAC

Часть II-ая. И не последняя :)))

Итак в первой части я рассмотрел как "привязать" мультибитные ЦАПы с последовательной загрузкой данных к популярному ресиверу CS8412\14 посредством прерывания тактовой частоты загрузки данных в регистр ЦАПа.

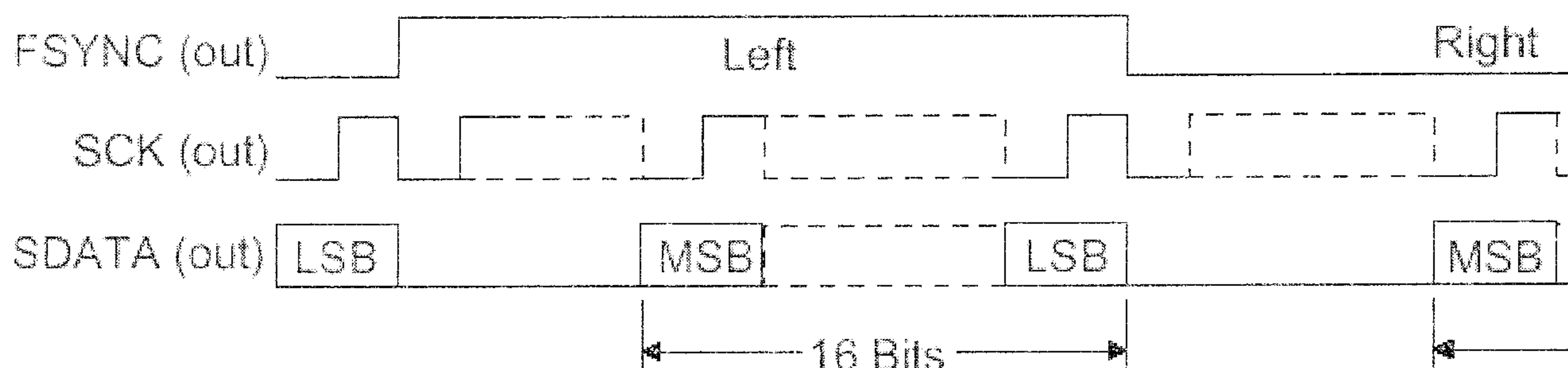
К сожалению не все м\с ЦАП допускают использование такого способа. Да и не всем хорош такой способ. Например, отказались работать с прерыванием тактовой такие прекрасные м\с, как PCM1704. Слишком уж требовательны оказались они к временным соотношениям сигналов на своих входах. Поэтому я публикую самый правильный способ соединения:

Способ второй: "сдвиговый регистр".

Пусть имеем: приемник цифрового потока CS8412(14) и пару м\с цап PCM56. У 8412(14) имеем три "ноги" SCK, FSYNC, SDATA. У PCM56 три "ноги": BCK, LE, DATA.

Начинаем с м\с ЦАП - данные отсчета (DATA) загружаются во внутренний регистр ЦАП по тактам BCK, после того как данные будут загружены в этот регистр, перепад уровня на "ноге" управления обновлением ЦАП (LE) заставит ЦАП выдать ток на аналоговом выходе, соответствующий цифровому отсчету и удерживать его до следующего обновления.

Ниже приведена диаграмма выходных сигналов ресивера CS8412 в режиме "5":



Первым следуют данные левого канала, затем правого. В каждом полупериоде сигнала FSYNC уместается 32 периода сигнала SCK. Данные смещены к правой части полупериодов FSYNC (Right-Justified).

Мы знаем (из даташита) что данные грузятся в регистр м\с цап по положительному перепаду сигнала CLK (он же SCK). Также мы видим, что обновление (Latch) м\с цап мы должны производить перепадом от высокого уровня к низкому ("отрицательный" перепад) сигнала LE (Latch Enable) после загрузки данных левого и правого каналов.

Исходя из вышеперечисленного формулируем задачу:

- 16-бит данных отсчета левого канала должны грузиться в м\с цап одновременно с данными правого канала
- сигналом LE для м\с цап должен стать инвертированный сигнал FSYNC

Решение:

- загрузив поток данных в 32-разрядный сдвиговый регистр, тактируемый сигналом SCK, мы получим на выходе этого регистра данные левого канала в тот момент когда на самой линии SDATA будут идти данные правого канала
- сигнал FSYNC проинвертируем и направим на выводы LE м\с цап

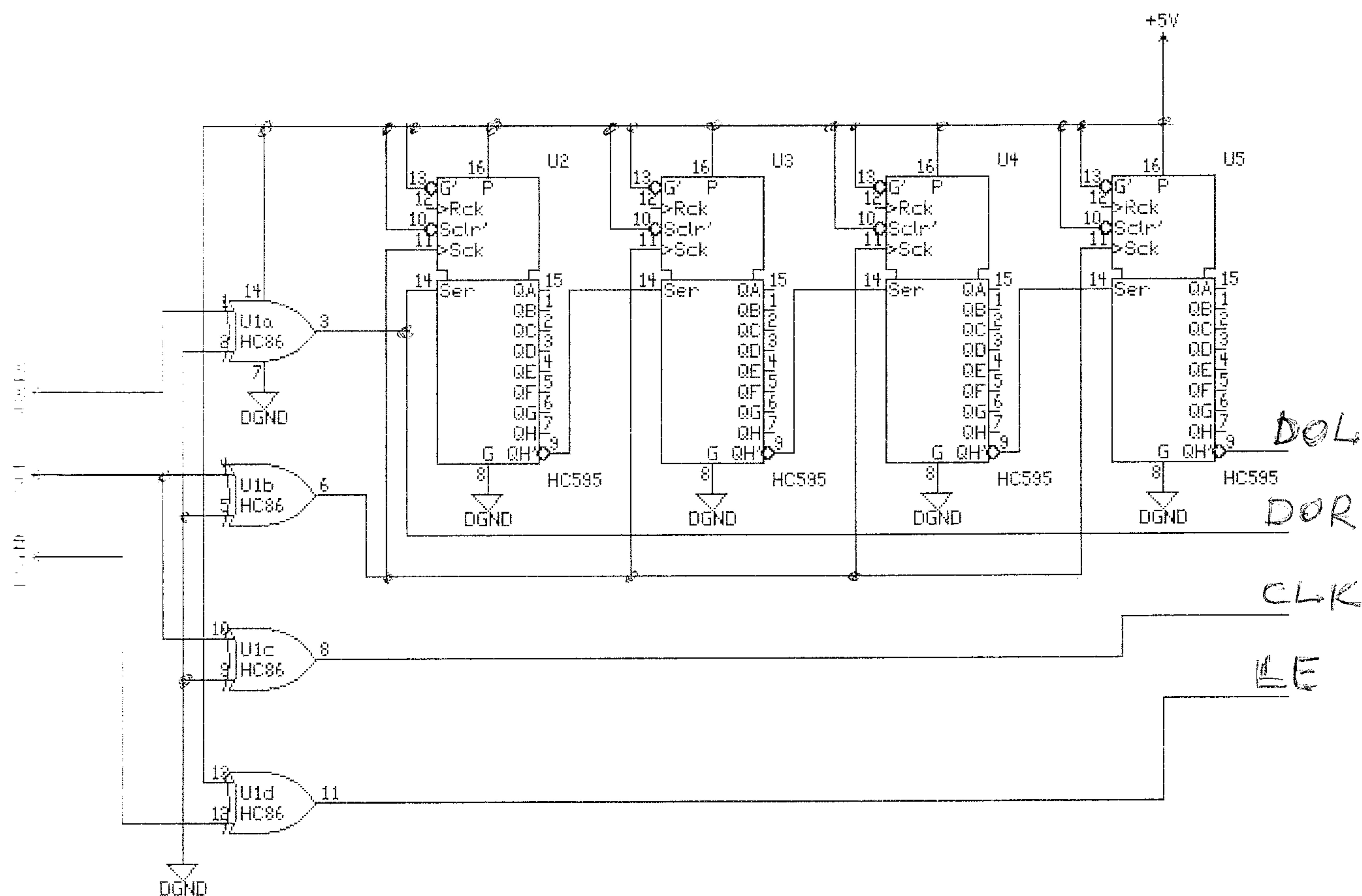
Выбираем регистр сдвига, тонкости.

Первое, о чем надо помнить, это то, что данные должны загружаться и выгружаться из регистра по положительному перепаду сигнала SCK, также как и у внутреннего регистра цап. Второе - быстродействие регистра должно быть не ниже 64Fs, а лучше больше раза в три-четыре.

Готового на 32-разряда я не нашел. Чаще всего под рукой оказываются 8-разрядные сдвиговые регистры. У меня, например, были 74НС595, работающие вплоть до частот 100Мгц, хотя можно сделать и на 299-х регистрах. Надобно таких регистров 4 шт. ($4 \times 8 = 32$).

В качестве буферных элементов применена м\с 74НС86. Ее назначение - "разгрузить" выходы ресивера, а также при необходимости легко заставить ее инвертировать проходящий сигнал.

Ну вот и собственно сама схема:



Сигналы обозначены так, как они обычно обозначаются на готовых схемах цап с циф. фильтрами. Вот собственно и все. Творите!

А в третьей части я расскажу как сделать этот узел цап без цифрового фильтра если источником цифровых сигналов выступает шина I2S. Но не пугайтесь! 30-40 корпусов логики или какой-нибудь заказной ИС там не будет. Все гораздо проще :)))

SE-6

Относительно недавно увидел, что одна известная фирма :), не скажу какая в довершение ко всему стала выпускать помимо внешних ЦАП без передискретизации и цифрового фильтра еще и проигрыватели компакт-дисков также без вышеозначенных штук. Стоимость как всегда немаленькая (по нашим российским меркам).

На самом деле многие "обычные" проигрыватели и CD-ROM приводы легко превратить в устройства, аналогичные "фирменным". Пусть и нельзя будет достичь "Высокого конца", зато радость творчества и результат могут быть полезны и интересны для творящего.

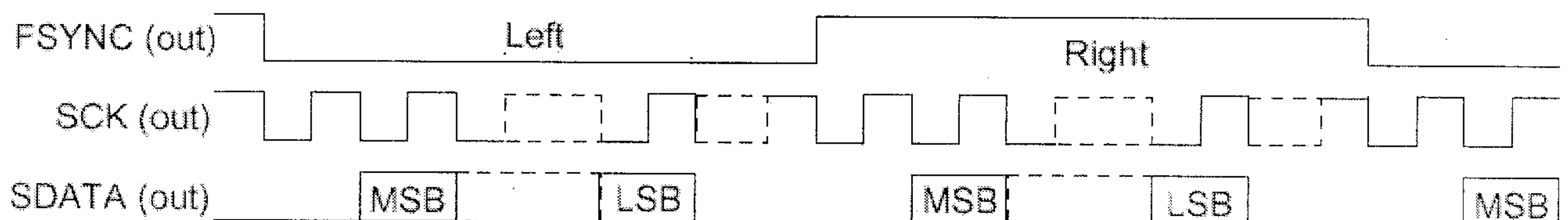
Способ третий. Эта "ужасная" I2S.

Перво-наперво хочется внести некоторую ясность в описываемый предмет - есть шина I2S, а есть формат передачи данных по этой шине. Не стоит смешивать эти понятия.

Итак шина I2S, как помнится состоит из трех сигналов:

- BCK (он же CLK, он же SCK, он же... да не суть важно) - сигнал, по которому тактируется прием данных регистром ЦАПа
- LRCK (LRCLK, FSYNC...) - сигнал, обозначающий данные, какого канала в данный момент передаются по шине.
- DATA - собственно данные отсчетов каналов

На рисунке приведен формат передачи данных I2S по шине I2S ;)



Что видно из этой диаграммы:

Данные отсчетов каналов смещены на один "клок" сигнала SCK (BCK, CLK) относительно перепадов сигнала FSYNC (LRCLK, FSYNC...).

Чаще всего в одном полупериоде сигнала LRCK укладывается 32 "клока" сигнала BCK. Собственно вся проблема состоит только в том, чтобы превратить исходный формат в формат Right-Justified посредством... правильно, сдвигового регистра (мне кажется это проще, хотя можно использовать и другие способы - с подсчетом "клоков", например, но выигрыша в количестве логики мы вряд ли получим).

Считаем: 32 "клока" в периоде, смещение на один клок есть, пусть наши ЦАПы 16 битные и мы предполагаем, что поток данных также 16-битный (чаще всего так и есть, но не всегда). Стало-быть "придержав" поток данных на $32-16-1=15$ "клоков" мы превратим формат I2S в формат Right-Justified, причем если далее у нас используется 32-х разрядный сдвиговый регистр, то можно просто "придержать" всю последовательность данных одним единственным сдвиговым регистром, включенным до 32-х разрядного. А используя получившийся 48 (32+16) разрядный сдвиговый регистр как регистр с переменной длиной можно стыковать цапы с большей разрядностью и шину с меньшей (но не наоборот!).

Схема универсального сдвигового регистра для практически всех случаев в жизни ;) -

Сигналы обозначены так, как они обычно обозначаются на готовых схемах циф. фильтрами. Вот собственно и все. Конечная конфигурация определяется переключателями. Переключки у буферов сигналов DATA и LRCK требуются для инвертирования этих сигналов, что иногда бывает необходимо.

Картинки готового устройства и платы ЦАП выложу чуть позже.

Splog

