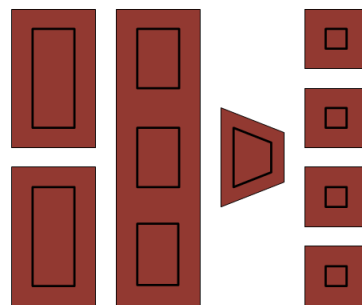


Сообщество разработчиков на FPGA и SoC



FPGA-Systems.ru



SPI :: Часть 2:: Варианты реализации

Автор: KeisN13



Оглавление

Доступные ресурсы ПЛИС.....	3
Варианты реализации интерфейса SPI.....	4
Вариант №1: Конечный автомат.....	4
Вариант №2: Память	4
Вариант №3: Сдвиговый регистр	5
Вариант №4: Счётчик	6
Преобразование данных с АЦП.....	6
Архитектура проекта	7
Аппаратное ограничение	8
Окончательная архитектура проекта.....	8
Заключение	9

Здравствуйте друзья. С вами проект FPGA-systems.ru.

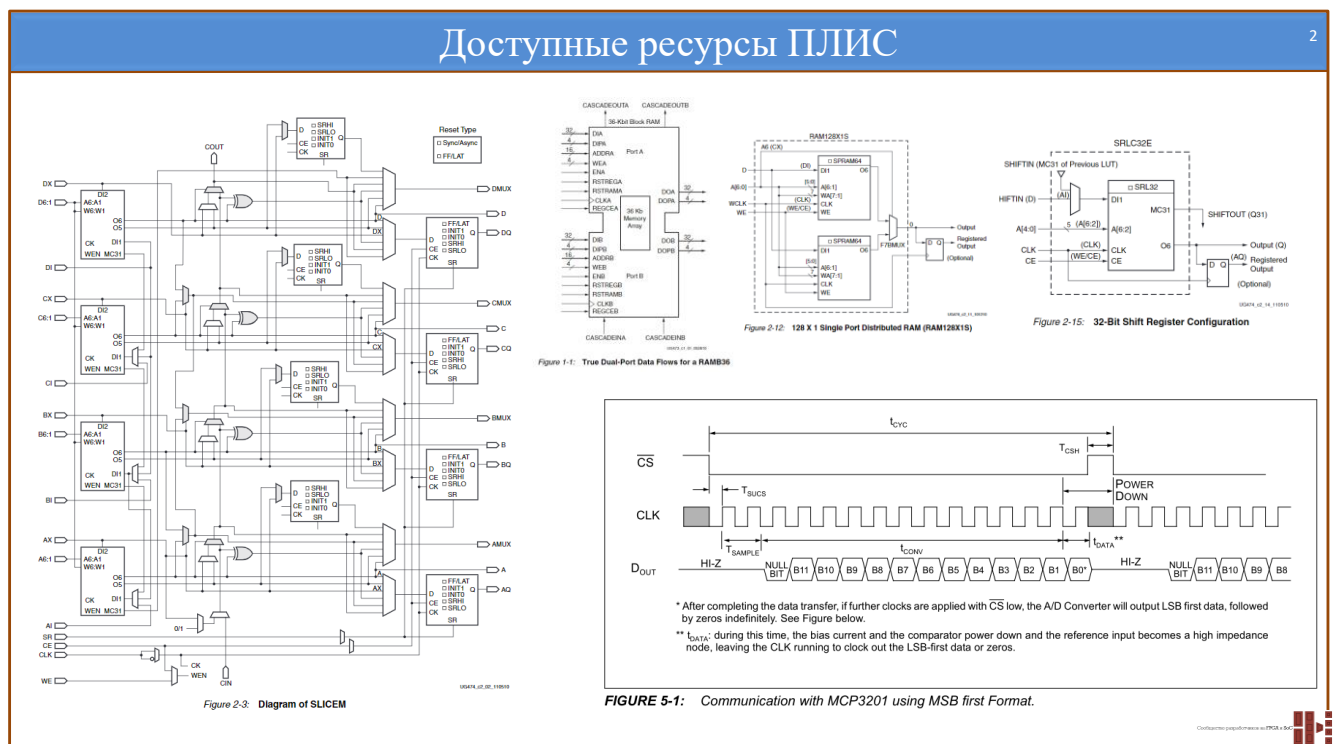
Продолжаем работу над проектом по развёртыванию интерфейса SPI на ПЛИС. Сегодня мы с Вами обсудим варианты реализации интерфейса.

Доступные ресурсы ПЛИС

Внутри ПЛИС имеется несколько базовых компонентов, которые могут быть использованы для организации интерфейса SPI, что приводит нас к тому, что нам следует определиться каким именно способом мы будем реализовывать интерфейс и какие элементы ПЛИС при этом мы будем задействовать.

Итак, давайте для начала изучим документацию на выбранный нами ранее АЦП и посмотрим, какие варианты построения интерфейса могут быть для него использованы. Напомню, что в первой части мы выбрали недорогой АЦП с интерфейсом SPI: MCP3201.

Обратимся непосредственно к временной диаграмме интерфейса. Мы с Вами видим, что АЦП управляется двумя сигналами: Chip Select и CLK. Отсчёты оцифрованного аналогового сигнала выходят через выход Dout АЦП в последовательном виде в формате MSB, то есть первым передается старший байт.



Наша задача состоит в том, что бы эту временную диаграмму повторить, соблюдая при этом все указанные на ней временные интервалы.

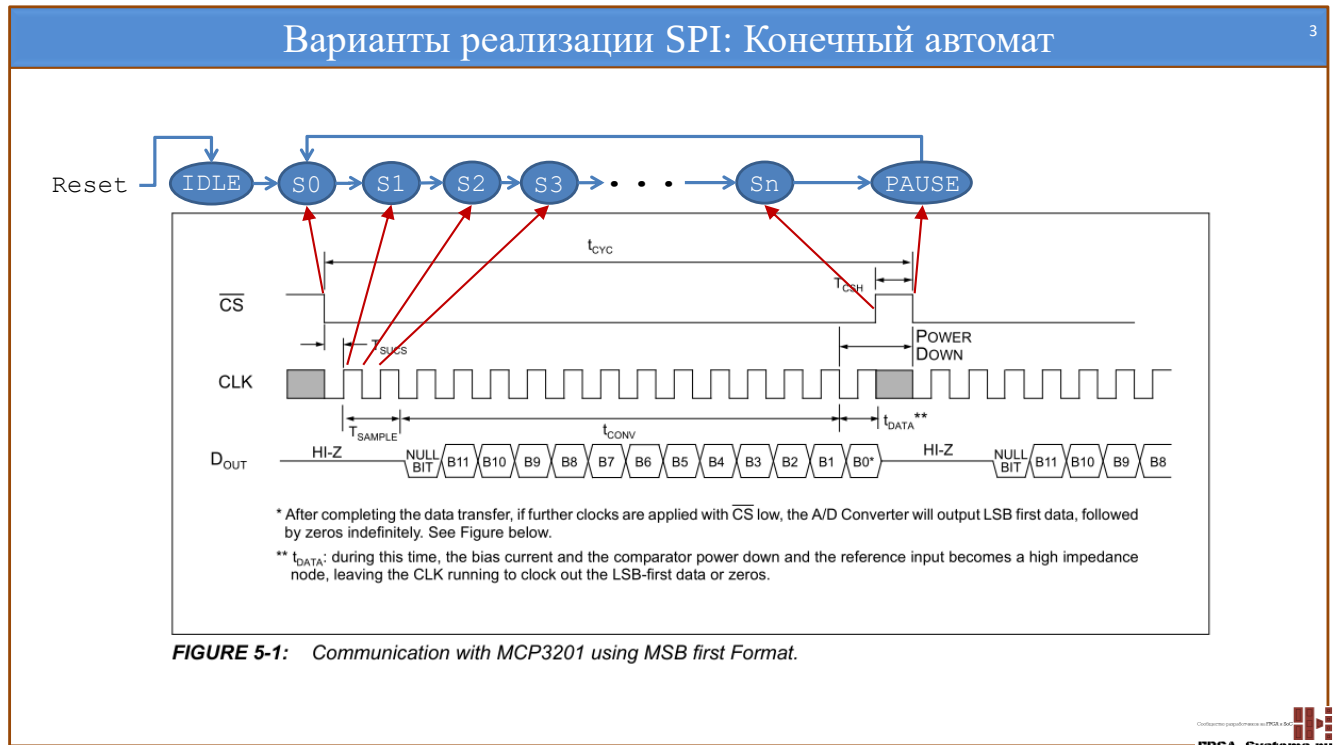
Давайте теперь подумаем, а какие у нас есть инструменты и средства, чтобы это сделать.

Существует множество вариантов формирования управляющих сигналов, которые ограничены лишь Вашей фантазией. Приведем некоторые из них.

Варианты реализации интерфейса SPI

Вариант №1: Конечный автомат

Первый вариант может быть основан на построении конечного автомата, в состояниях которого формируются соответствующие сигналы управления.



Такой подход позволит нам пошагово воспроизвести временную диаграмму, и при этом потребует от нас знаний описания конечного автомата как такового, и, возможно, некоторых более глубоких знаний. Например, стилей описания конечного автомата: один, двух или трех процессный, задания вариантов кодирования состояния конечного автомата: binary, gray, one-hot и др. Но с большой долей вероятности данные знания нам не пригодятся, и мы сможем без проблем разработать конечный автомат.

Вариант №2: Память

Поскольку наши сигналы управления задаются в виде определенно заданной последовательности, которая не меняется от отсчёта к отсчёту, то мы можем создать память, в которую будут занесены значения управляющих сигналов. Используя лишь только тактовый сигнал и счетчик адреса мы сможем последовательно формировать состояния управляющих сигналов. В виду малого размера последовательности мы можем разместить ее в распределённой памяти ПЛИС, а не в блочной, что было бы достаточно расточительно.

Варианты реализации SPI: Память

4

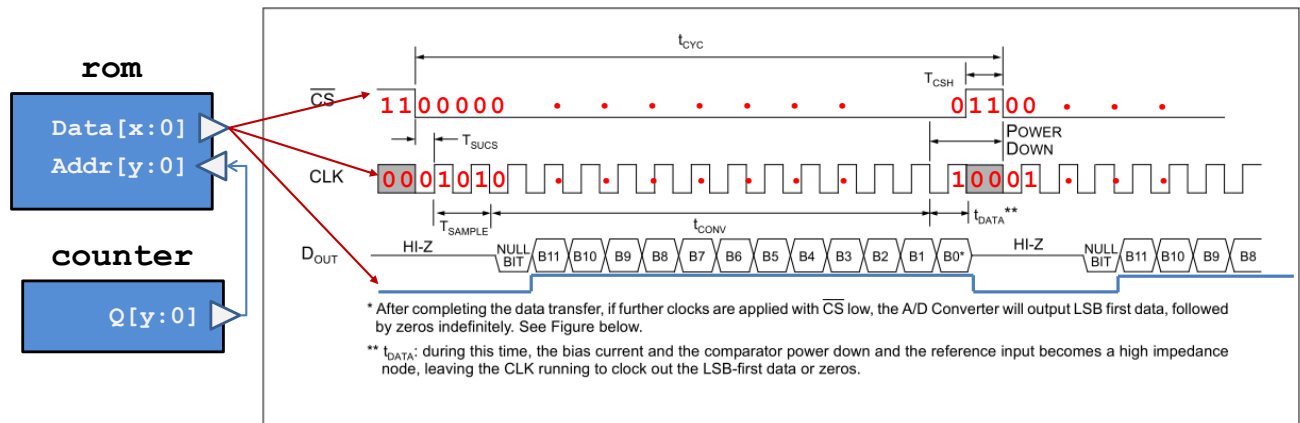


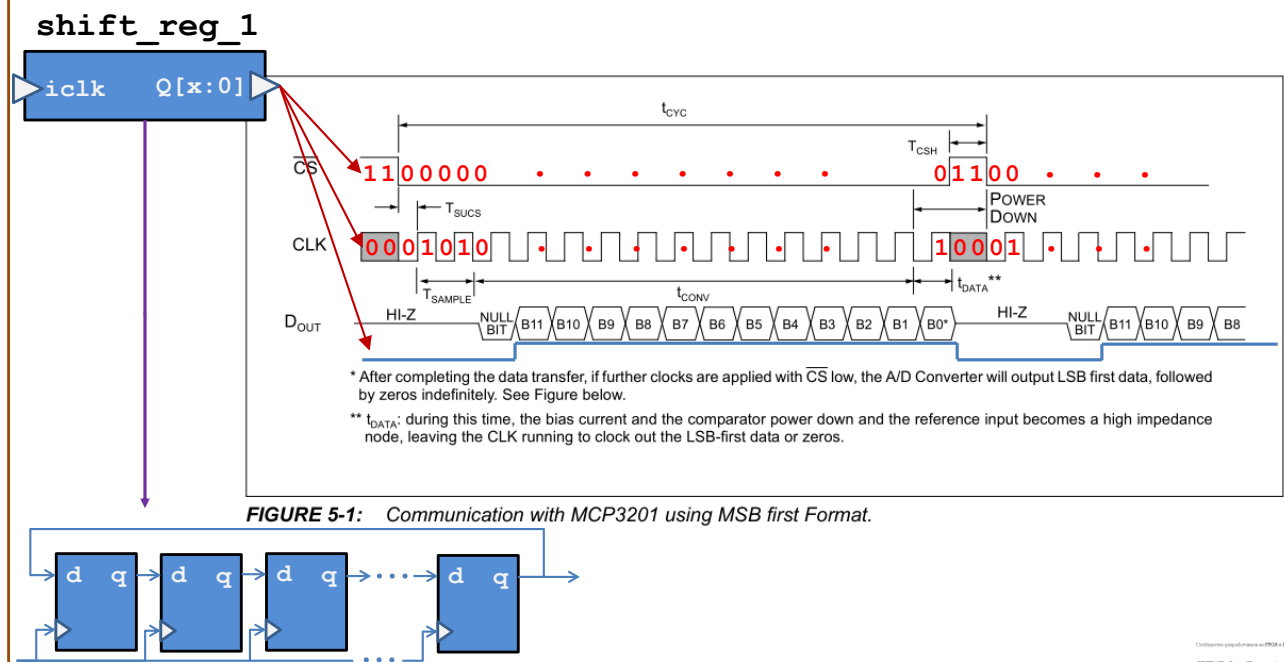
FIGURE 5-1: Communication with MCP3201 using MSB first Format.

Вариант №3: Сдвиговый регистр

Также вместо памяти и счетчика адреса мы можем использовать зацикленный сдвиговый регистр, значения которого инициализируются необходимой нам последовательностью. В таком подходе нам не нужно формировать значение адреса, поскольку нет памяти, а последовательность циклически повторяется и перезаписывает сама себя. Сдвиговый регистр можно выполнить как на отдельно взятых триггерах, так и на специализированных ресурсах SRL, что позволит сделать разрабатываемый модуль крайне компактным

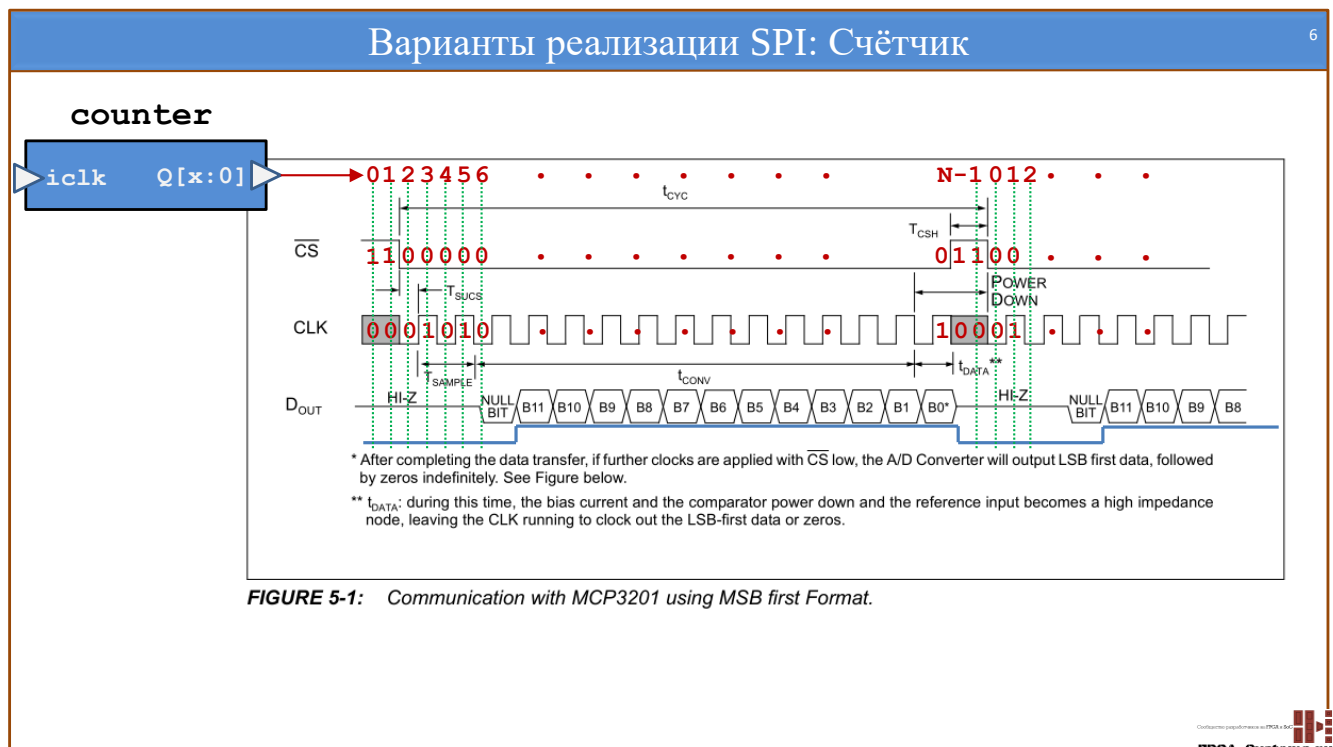
Варианты реализации: Сдвиговый регистр

5



Вариант №4: Счётчик

Мы можем попробовать использовать простой счётчик, значения сигналов управления мы сможем контролировать, основываясь на значениях счётчика.



Мы будем разрабатывать код для определённого АЦП и не будем делать наш модуль SPI максимально универсальным, поскольку это потребует более глубоких знаний интерфейса SPI. К тому же разные производители могут изменять временные характеристики интерфейса, количество передаваемых данных, время отклика и т.д. В виду такого большого разброса параметров гораздо удобнее будет сделать интерфейс под конкретную микросхему, в нашем случае это MCP3201. К тому же это не займет много времени да и как часто Вам приходится менять один АЦП на другой?

Предлагаемые решения вполне универсальны и позволят Вам без особых временных затрат переделать модуль под другие периферийные устройства с интерфейсом SPI.

Преобразование данных с АЦП

Преобразование последовательного формата значения отсчёта оцифрованного сигнала в параллельный мы возложим на обычный сдвиговый регистр. В виду того, что данные с АЦП приходят не сразу, а имеют пустой бит и определённую временную задержку, мы сформируем дополнительный сигнал разрешения, который будет являться комбинацией сигналов CLK и дополнительного сигнала валидности данных с АЦП (показанного синей линией), который мы также сформируем самостоятельно. Не стоит забывать и о сигнале готовности преобразования последовательного формата значения отсчёта в параллельный.

Преобразование данных с АЦП

7

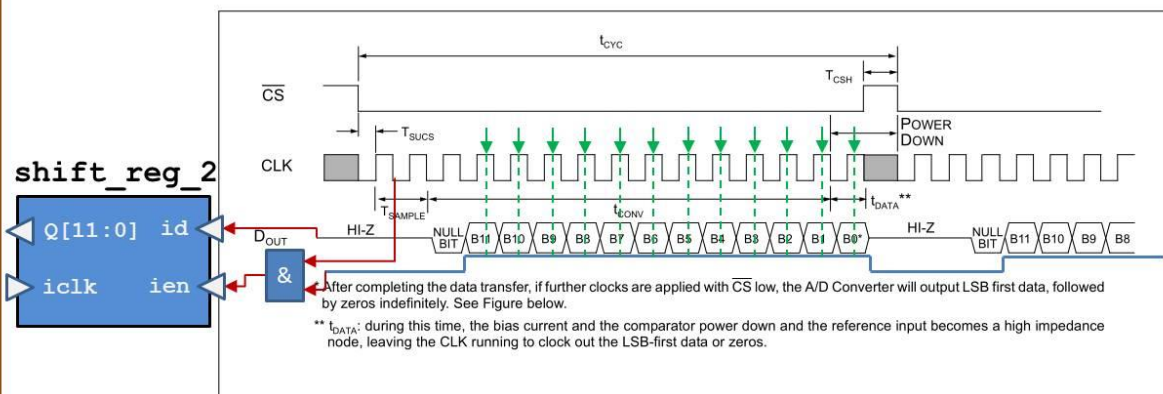


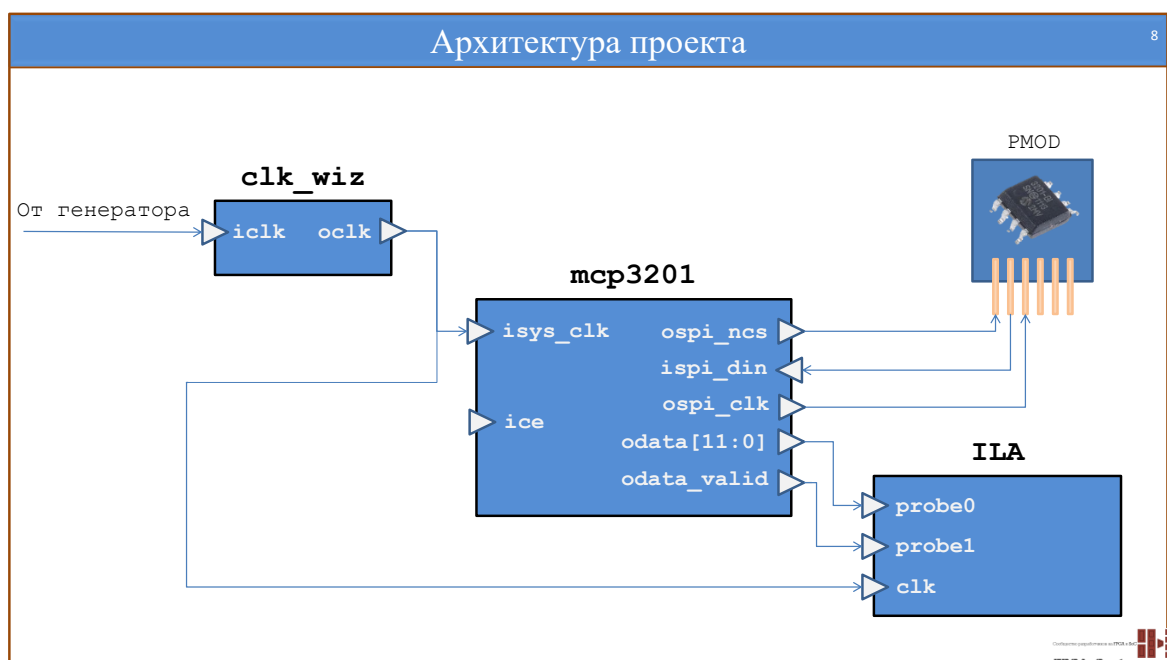
FIGURE 5-1: Communication with MCP3201 using MSB first Format.

Архитектура проекта

Подводя итог вышесказанному, продумаем архитектуру проекта. Она должна содержать:

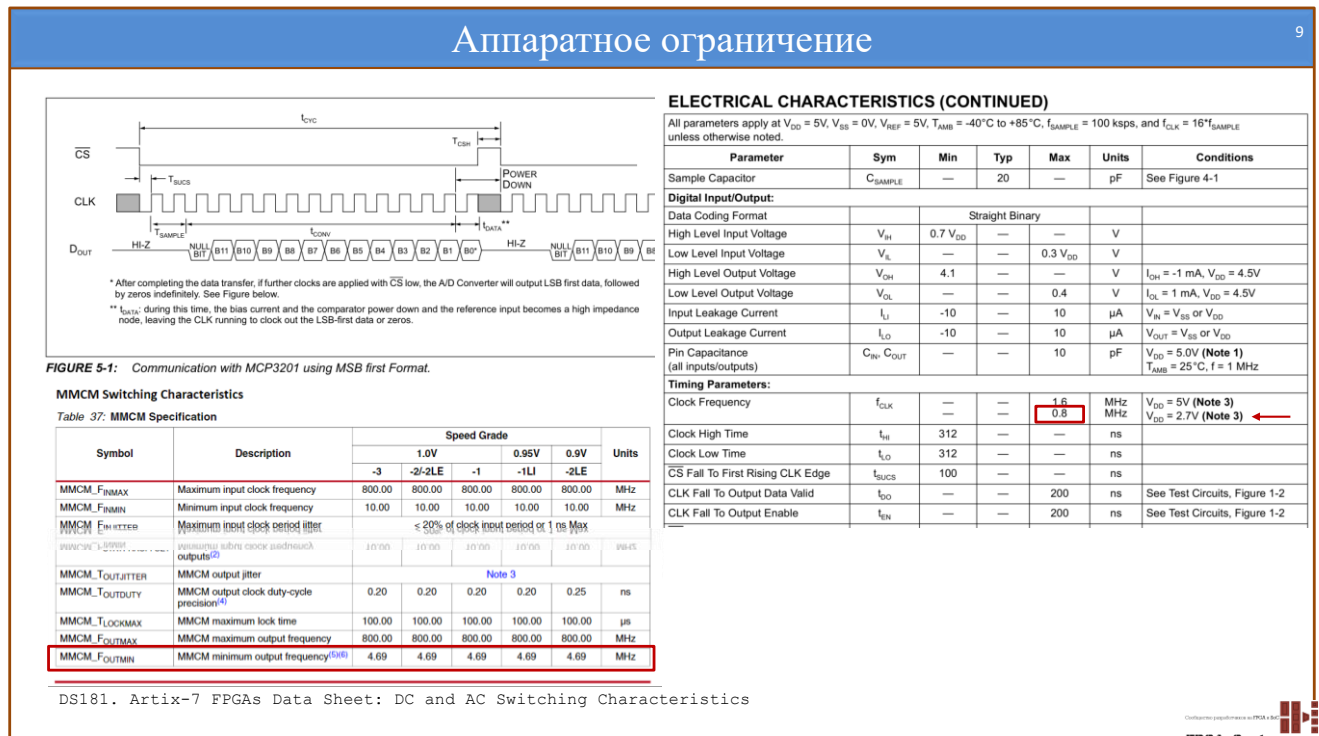
- формирователь системной тактовой частоты `clk_wiz`
- модуль управления АЦП `mcp3201` (название модуля соответствует подключаемому АЦП)
- интегрированный логический анализатор ИЛА, который будет собирать считанные данные с АЦП в некоторый массив. Затем этот массив будет передан по JTAG в на компьютер в Vivado для его отображения на экране.

Однако следует учесть еще один не маловажный фактор. Он состоит в значении рабочей частоты АЦП и частоты, которую может сгенерировать модуль `clk_wiz`.



Аппаратное ограничение

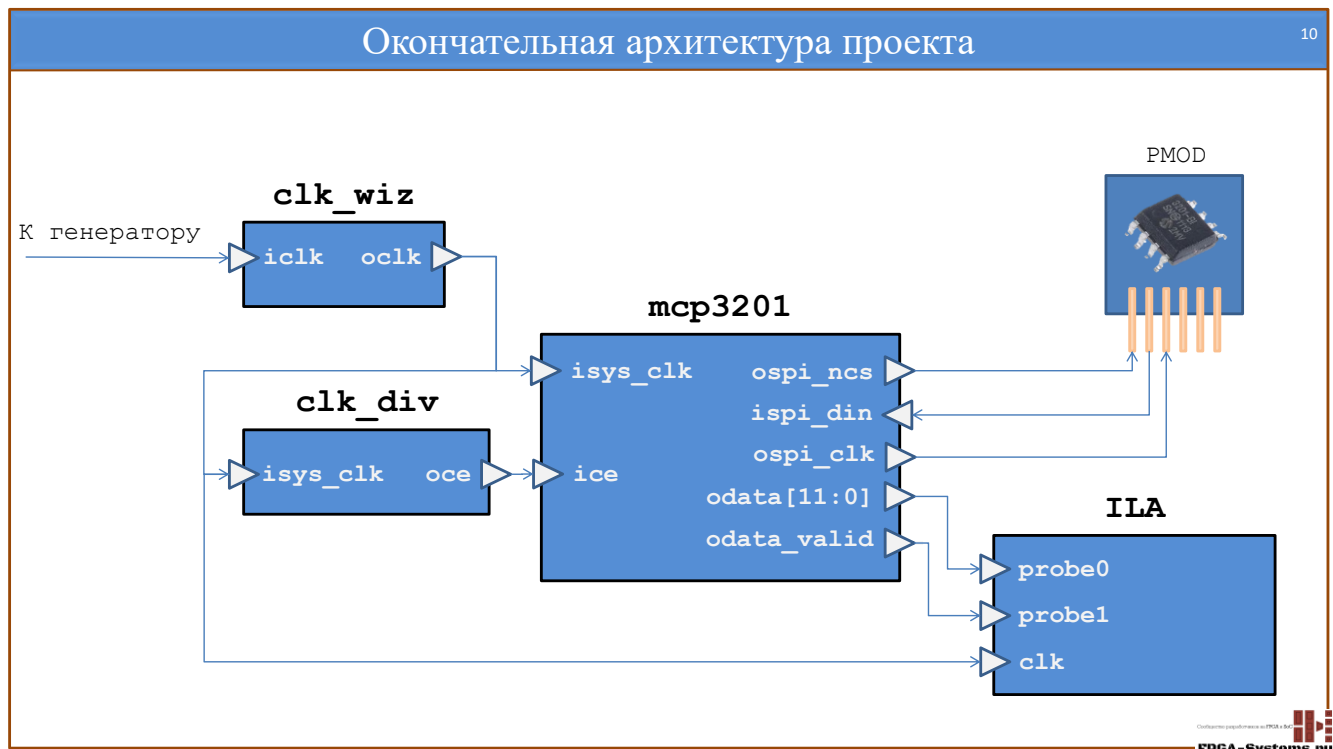
Обратимся вновь к временной диаграмме и таблице параметров АЦП. Как мы видим указаны значения для напряжения питания 5В и 2.7В. Давайте будем ориентироваться на параметры при питании 2.7В, поскольку напряжение получаемое с PMOD равно 3.3В. При таком напряжении питания, частота следования тактовых импульсов CLK не должна превышать 0.8 МГц. Это в свою очередь приводит нас к тому, что мы будем вынуждены поставить блок деления системной тактовой частоты.



Его наличие будет необходимо нам по той причине, что PLL или MMCM 7-го семейства ПЛИС Xilinx имеют минимальную частоту, которую они смогут сгенерировать. Согласно документации на ПЛИС (DS181) она составляет примерно 5МГц. Устанавливая этот модуль и задавая необходимый для нас коэффициент деления системной тактовой частоты, мы добьемся корректных значений временных интервалов для формируемых сигналов управления АЦП.

Окончательная архитектура проекта

Теперь полная схема проекта и подключения модулей будет выглядеть следующим образом.



Сам модуль управления АЦП mcp3201 имеет следующие порты ввода/вывода:

- **isys_clk** – системный тактовый сигнал, генерируемый модулем clk_wiz
- **ice** – вход разрешения системной тактовой частоты
- **ispi_din** - вход данных от АЦП
- **ospi_clk** – тактовый сигнал интерфейса SPI для АЦП
- **ospi_ncs** – сигнал управления chip select
- **odata[11 : 0]** – 12битное значение отсчёта сигнала, оцифрованного АЦП
- **odata_valid** – сигнал валидности принятого отсчёта

Заключение

На этом наше обсуждение вариантов реализации и архитектуры проектов закончено. В следующей части мы приступим к разработке кода на языках VHDL и Verilog. А для того, чтобы не пропустить выход следующего видео или статьи, не забудьте подписаться на наш канал и присоединиться к нам в соц сетях и чату в telegram.

Также нам интересно узнать, какие варианты реализации Вам интересны или какие Вы уже использовали на практике. Не забудьте написать об этом в комментариях.

До встречи в следующей части.

У Вас есть вопрос, идея или
«я хочу помочь в развитии проекта статейкой, видео, поделиться FPGA-шным опытом»

Оставляйте Ваши вопросы, идеи и предложения в комментариях или на
www.fpga-systems.ru или присылайте их на fpga-systems@yandex.ru



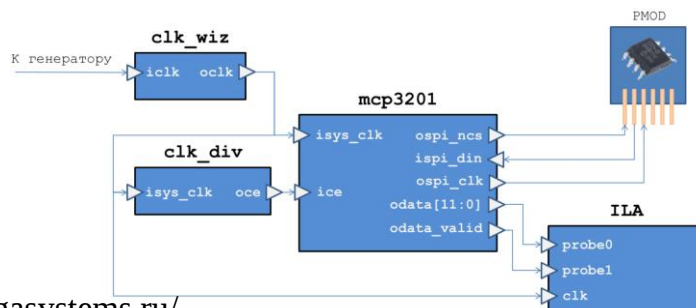
@Powered_by_KeisN13



vk.com/club185679360



www.facebook.com/groups/fpgasystems.ru/



Слайды презентации доступны на сайте проекта www.fpga-systems.ru

Не забудьте присоединиться к нам в соцсетях



и

зарегистрироваться на сайте, чтобы не пропустить выход новых видео и уроков