Практика использования встроенного АЦП в ПЛИС семейства MAX10

Часть 3. Цифровой вольтметр/термометр на базе АЦП ПЛИС MAX10

Павел Редькин (г. Ульяновск)

Предлагаемая статья содержит информацию по практическому применению аппаратного модуля АЦП, входящего в состав ПЛИС семейства MAX10 производства Intel (Altera). В третьей части статьи изложенная в предыдущих частях справочная информация иллюстрируется на примере рабочего проекта ПЛИС с АЦП, реализующего функции цифрового вольтметра и цифрового термометра с выводом результатов измерений на индикатор и во внешние устройства.

Функции проекта ПЛИС

Описанный в статье проект ПЛИС, реализующий обслуживание IP-ядра модуля АЦП в ПЛИС, является тестовым. Он разработан для ПЛИС 10M08SAE144C8G из состава отладочной платы Altera MAX10 FPGA. Проект предназначен для исследования АЦП, а также для отладки взаимодействия модуля АЦП в ПЛИС и цифровой части ПЛИС и внешних по отношению к ПЛИС устройств. Вместе с тем этот проект вполне успешно реализует функции цифрового вольтметра и цифрового термометра, используя для реализации последней функции встроенный температурный датчик модуля АЦП в ПЛИС.

Проект ПЛИС в виде архивного файла MAX_10_ADC_1.zip можно загрузить с сайта журнала.

Проект ПЛИС позволяет производить исследование любого из каналов АЦП, однако аппаратно для исследования АЦП больше всего подходит его канал 7, входные цепи которого на отладочной плате Altera MAX10 FPGA снабжены потенциометром для обеспечения возможности регулировки входного измеряемого напряжения АЦП.

В качестве входного измеряемого напряжения для АЦП в ПЛИС можно использовать внешнее постоянное напряжение в диапазоне от 0 до +3,3 В относительно аналогового общего провода (обозначение цепи ⏚ на схеме рисунка 10 во 2-й части статьи), поступающее на вход повторителя напряжения на операционном усилителе (ОУ) U4C с контакта 7 разъёма J4 (цепь Arduino_A6 на схеме рисунка 10). С помощью перемычки в джампере J7 можно альтернативно подать на вход ОУ регулируемое напряжение с движка потенциометра R16. Выход ОУ U4C через фильтрующую RC-цепочку R92, C59 подключён к универсальному входу АЦП в ПЛИС ADCIN7 (канал 7 АЦП). Резистор R84, образующий нижнее плечо предварительного делителя входного напряжения на 2, автор в ходе экспериментов с АЦП из отладочной платы демонтировал. Кроме того, в отладочную плату были внесены следующие доработки:

- удалены элементы R73, C46, подключённые к выделенному входу АЦП ANAIN1 (выводу 3 ПЛИС U2);
- выделенный вход АЦП ANAIN1 (вывод 3 ПЛИС U2) соединён методом навесного монтажа с цепью Arduino_A6 в точке контакта 1 джампера J7.

Проведённые доработки обеспечивают возможность использования в проекте выделенного входа ПЛИС.

Для «ручного» управления режимом и состоянием АЦП в собранном на основе отладочной платы макете предусмотрены кнопки управления SB1...SB6, которые позволяют выбирать канал АЦП для преобразований по кольцу и задавать уровни цифровых сигналов управления АЦП.

Все текущие параметры режима АЦП и результаты преобразований проект ПЛИС отображает на подключённом к ПЛИС ЖКИ: состояние АЦП (включено/отключено) «VAL_ADC=ON/OFF», запуск/останов АЦП «ADC=Pusk/Stop», выбранный для преобразований канал АЦП «CHAN=xx», где xx – номер канала в диапазоне от 0 до 20, текущий результат АЦП в битах и милливольтах «ADC_OUT=yyyy bit» и «U=zzzz mV», значение температуры кристалла ПЛИС со знаком в °C «T= mmmm C» в случае, если для преобразований выбран канал встроенного температурного датчика TSD (канал 17). Обновление показаний ЖКИ осуществляется каждые 0,5 с. Никакого усреднения результатов АЦП для вывода на индикацию в проекте ПЛИС не производится: на индикацию поступает просто каждый пятисоттысячный результат при измерении внешнего напряжения или каждый двадцатипятитысячный – при измерении температуры.

Текущее состояние сигналов command_startofpacket_ADC и command_endofpacket_ADC, с помощью которых осуществляются запуск и останов АЦП, помимо индикации на ЖКИ («ADC=Pusk/Stop») дублируется штатными светодиодами отладочной платы D5 и D4 соответственно. Свечение светодиода указывает на высокий (активный) уровень соответствующего сигнала. Светодиод D1 используется для индикации секундного ритма.

Для выдачи результатов АЦП в проекте ПЛИС одновременно используются три способа: индикация результата АЦП на ЖКИ в битах, милливольтах и °C (последнее только в случае выбора для преобразований канала 17 температурного датчика TSD), выдача результата АЦП из ПЛИС в параллельном виде в битах, выдача результата АЦП из ПЛИС в последовательном виде в битах. Выдача результатов АЦП в параллельном и последовательном виде реализована для обеспечения возможности детального, в том числе статистического, исследования АЦП, поскольку через эти интерфейсы выдаётся результат каждой производимой АЦП выборки. Индикация результата АЦП на ЖКИ реализована в проекте ПЛИС для обеспечения общей визуальной оценки АЦП.

Структура проекта ПЛИС

Исходные коды проекта ПЛИС написаны на языке описания аппаратных

средств Verilog HDL. Структурно проект ПЛИС состоит из исходного файла модуля верхнего уровня MAX_10_ADC_1.v и нескольких исходных файлов модулей более низкого уровня: файла модуля обслуживания ЖКИ 12864ZW LCD_12864.v, файла модуля формирования единичного импульса разрешения звука от нажатия кнопки buzzer_butt.v, файла модуля фильтра дребезга контактов кнопки noise_filter_butt.v, файла модуля выдачи результата АЦП в последовательном виде ADC_rezult_serial.v, файла модуля преобразования выходного кода АЦП в значение температуры в °С со знаком Code_Temper_conv.v, а также двух файлов, сгенерированных в ходе создания проекта инструментом Qsys: файла IP-ядра модуля АЦП ADC_Core_1.qip, файла IP-ядра модуля PLL ALTPLL1.qip, каждый с исходными файлами более низкого уровня. Полная файловая структура проекта ПЛИС показана на рисунке 24 (2-я часть статьи). Проект занимает около 1/3 ресурсов ПЛИС 10M08SAE144C8G по логике и около 1/2 – по линиям ввода-вывода.

В модуле верхнего уровня реализована главная функция проекта ПЛИС – осуществление аналого-цифровых преобразований, а также несколько сервисных функций: опрос кнопок управления, задание значений управляющих сигналов АЦП по результатам этого опроса, вывод текущих настроек АЦП и результатов преобразований на ЖКИ, вывод результатов преобразований через цифровые интерфейсы, генерация внутренних синхросигналов, генерация звуковых сигналов, управление светодиодами.

Выбор канала АЦП для преобразований осуществляется кнопкой SB4 по кольцу от 0 до 20.

Кнопкой SB1 можно управлять значением сигнала command_valid_ADC по кольцу, то есть каждое нажатие на кнопку приводит к инверсии текущего значения сигнала. Сигнал command_valid_ADC, по сути, задаёт состояние АЦП: включено или отключено. Опытным путём было установлено, что выбор канала для преобразований всегда нужно производить при отключённом АЦП, то есть при низком уровне сигнала command_valid_ADC. В противном случае возможны сбои в установлении частоты синхросигнала АЦП. Так, вместо синхросигнала с частотой 50 кГц, автоматически генерируемого IP-ядром АЦП при выборе канала встроенного температурного датчика TSD, может ошибочно генерироваться синхросигнал с большей в несколько раз частотой.

Листинг 5

```
// Фиксация результата АЦП по фронту сигнала
// готовности результата АЦП  command_ready_ADC
// при условии активного уровня сигнала
// валидности из АЦП response_valid_ADC
always @(posedge command_ready_ADC)
begin
  if (response_valid_ADC)
        begin
                  rezult_ADC <= dout_ADC;
        end
end
```

Листинг 6

```
parameter  U_ref_mV = 3300;  // опорное напряжение АЦП в милливольтах
                             // для пересчета результата АЦП в мВ
// умножение результата АЦП в битах на опорное напряжение в милливольтах,
    // чтобы получить результат в милливольтах
     rezult_ADC_U_ref <= rezult_ADC * U_ref_mV;

    // деление результата умножения на  2**12,
    // что эквивалентно лог сдвигу вправо на 12 разрядов
         rezult_ADC_mV <= (rezult_ADC_U_ref >> 12);

     rezult_ADC_mV_tis <= rezult_ADC_mV / 1000;
     Temp2_data <= rezult_ADC_mV % 1000;
     rezult_ADC_mV_sot <= Temp2_data / 100;
     Temp3_data <= Temp2_data % 100;
     rezult_ADC_mV_des <= Temp3_data / 10;
     rezult_ADC_mV_ed <= Temp3_data % 10;
```

Кнопки SB2, SB3 управляют по кольцу значениями сигналов command_startofpacket_ADC, command_endofpacket_ADC соответственно. Однако поскольку в проекте ПЛИС мы используем IP-ядро АЦП без программы упорядочения (sequencer), значения указанных сигналов не должны никак влиять на ход преобразований. В нашем проекте преобразования автоматически запускаются только при установке сигнала command_valid_ADC и останавливаются только при сбросе этого сигнала. Значения сигналов command_startofpacket_ADC, command_endofpacket_ADC при этом могут быть произвольными, управление ими в текущей версии проекта ПЛИС реализовано для будущих версий.

Кнопки SB5, SB6 в текущей версии проекта ПЛИС не используются, они зарезервированы для будущих версий.

Все кнопки SB1-SB6 работают на замыкание между цифровым входом ПЛИС и цифровым общим проводом. В проекте ПЛИС задана подтяжка всех цифровых входов подключения кнопок к плюсу питания внутренними резисторами. Дребезг при нажатии на кнопки устраняется в проекте ПЛИС с помощью модуля фильтра дребезга контактов кнопки noise_filter_butt.v.

Аналого-цифровые преобразования при выборе любого канала, кроме канала встроенного температурного датчика TSD (канал 17), и установке сигнала command_valid_ADC осуществляются циклически, в автоматическом режиме с частотой выборок 1 МГц. При осуществлении выборок от TSD в канале 17 преобразования также осуществляются циклически, в автоматическом режиме, но уже с частотой 50 кГц. При осуществлении преобразований в любом канале фиксация результата АЦП производится по фронту сигнала готовности результата АЦП command_ready_ADC, поступающего из IP-ядра АЦП, при условии, что также поступающий из IP-ядра АЦП возвратный сигнал валидности результата АЦП response_valid_ADC находится на активном (высоком) уровне. Указанное действие реализуется с помощью языковой конструкции модуля, приведённой в листинге 5. В этом модуле dout_ADC – это 12-разрядная цепь (wire) выходов данных IP-ядра АЦП, а rezult_ADC – 12-разрядный регистр (reg) хранения текущего результата преобразования. Зафиксированное при каждом преобразовании в регистре rezult_ADC значение используется для дальнейшей обработки.

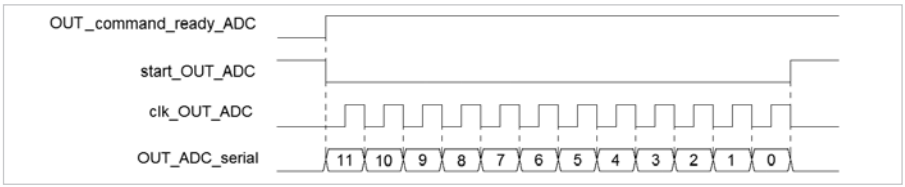

Рис. 29. Формат последовательной выдачи результата АЦП из ПЛИС

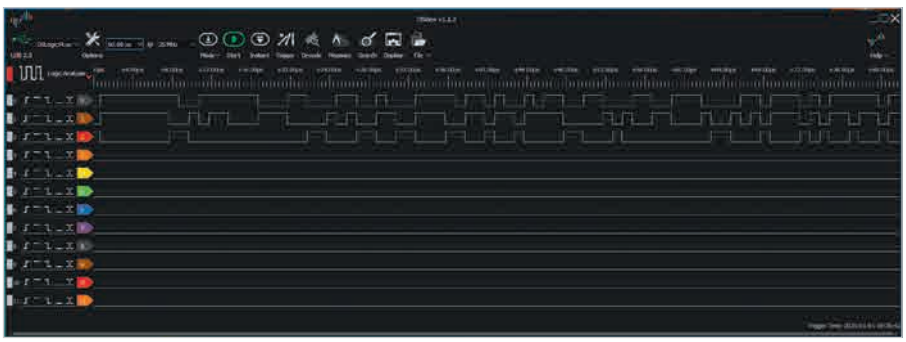

Рис. 30. Результаты АЦП для универсального входа ПЛИС на интервале 50 мкс для нулевого входного сигнала

Для пересчёта результата преобразования из битов в милливольты с последующим его преобразованием из двоичного в двоично-десятичное представление в проекте ПЛИС используется языковая конструкция, приведённая в листинге 6. В качестве исходных данных для пересчёта и преобразования представления здесь используется содержимое регистра rezult_ADC.

Для повышения точности преобразований значение parameter U_ref_mV, в авторском варианте проекта равное 3300, может быть скорректировано, исходя из конкретного значения напряжения опорного напряжения АЦП в милливольтах.

Выдача результата АЦП в последовательном виде во внешние устройства реализована с помощью модуля нижнего уровня, исходный текст которого содержится в файле ADC_rezult_serial.v. Формат последовательной выдачи в проекте ПЛИС в общих чертах воспроизводит распространённый формат SPI режима 0 (бит полярности SPOL = 0, бит комбинации фазы SPHA = 0). При этом ПЛИС является ведущим (master) устройством на шине и генерирует опорный синхросигнал. Сигнал принимаемых данных MISO в последовательном интерфейсе отсутствует, поскольку АЦП в ПЛИС не принимает извне никаких данных, сигналу передаваемых данных MOSI в проекте ПЛИС соответствует выход OUT_ADC_serial, сигналу синхронизации SCLK в проекте ПЛИС соответствует выход clk_OUT_ADC с частотой 25 МГц, сигналу выбора ведомого устройства /SS в проекте ПЛИС соответствует выход start_OUT_ADC. Однократная последовательная выдача 12-разрядного слова результата АЦП производится по каждому положительному фронту сигнала готовности результата АЦП OUT_command_ready_ADC, который представляет собой копию сигнала command_ready_ADC, выведенную из ПЛИС.

Временные диаграммы формата последовательной выдачи показаны на рисунке 29.

Параллельная выдача результата АЦП в проекте ПЛИС реализована в виде 12-разрядной параллельной выходной шины OUT_rezult_ADC_bus, слово данных на которой обновляется по каждому положительному фронту сигнала готовности результата АЦП OUT_command_ready_ADC. В качестве исходных данных для обновления шины здесь используется содержимое регистра rezult_ADC.

Преобразование выходного кода АЦП в значение температуры в °C со знаком реализована в проекте ПЛИС с помощью модуля нижнего уровня, исходный текст которого содержится в файле Code_Temper_conv.v. В основе модуля лежит таблица соответствия температуры выходному коду АЦП от температурного датчика TSD, взятая из [1, Table 4]. Каждому значению аргумента – 12-разрядного выходного кода АЦП code_ADC – в таблице сопоставлено пять 8-разрядных значений: двоичное значение модуля температуры в °C temper_mod_C, двоичное значение сотен модуля температуры в °C temper_mod_sot_C, двоичное значение десятков модуля температуры в °C temper_mod_des_C, двоичное значение единиц модуля температуры в °C temper_mod_ed_C, двоичное значение знака температуры temper_znak_C. Такой подход позволяет избежать необходимости дополнительного преобразования в проекте ПЛИС из двоичного в двоично-десятичное представление. Выдаваемые модулем Code_Temper_conv.v значения сотен, десятков и единиц модуля температуры и знака температуры поступают на индикацию после преобразования в модуле верхнего уровня в ASCII-формат, которое заключается в прибавлении к каждому байту десятичного числа 48.

Модуль обслуживания ЖКИ, исходный текст которого содержится в файле LCD_12864.v, сразу после включения питания ПЛИС производит аппаратный сброс подключённого к ПЛИС ЖКИ 12864ZW (длительность сброса 0,2 с, задаётся константой VALUE_END_RESET), начальную инициализацию ЖКИ, затем в течение 3 с на ЖКИ выводится начальная заставка, содержащая информацию о проекте ПЛИС и его версии. Время вывода начальной заставки задаётся константой VALUE_END_BOOT. После окончания индикации заставки начинает производиться циклический последовательный вывод на индикацию четырёх групп 8-разрядных регистров знакомест: data_LCD1_0_reg – data_LCD1_15_reg, data_LCD2_0_reg – data_LCD2_15_reg, data_LCD3_0_reg – data_LCD3_15_reg, data_LCD4_0_reg – data_LCD4_15_reg. Каждая группа включает по 16 регистров (по числу символов в строке ЖКИ) и соответствует одной строке ЖКИ (всего четыре строки: 1...4). Содержимое всех регистров обновляется одновременно в начале каждого цикла индикации содержимым соответствующих им 8-разрядных входов модуля обслуживания ЖКИ: data_LCD1_0 – data_LCD1_15, data_LCD2_0 – data_LCD2_15, data_LCD3_0 – data_LCD3_15, data_LCD4_0 – data_LCD4_15. На этих входах модуль верхнего уровня выставляет данные в произвольные моменты времени, поскольку разными входами в модуле верхнего уровня управляют разные цифровые автоматы. Цикл индикации (интервал обновления информации на ЖКИ) равен 0,5 с (задаётся константой VALUE_ZIKL). Синхронизация модуля обслуживания ЖКИ осуществляется внутренним синхросигналом с частотой 1кГц, поступающим из модуля верхнего уровня.

Помимо информационных сигналов проект ПЛИС выдаёт на выводы ПЛИС несколько контрольных и технологических сигналов: контрольный сигнал с выхода PLL sys_pll_control, сигнал возврата IP-ядром валидности АЦП OUT_response_valid_adc, сигнал возврата IP-ядром запуска преобразований OUT_response_startofpacket_adc, сигнал возврата IP-ядром останова преобразований OUT_response_endofpacket_adc.

Исследование измерителя напряжения в ПЛИС

Для исследования модуля АЦП с помощью вышеописанного проекта ПЛИС использовался многоканальный логический анализатор, описание которого можно загрузить по ссылке [2], а ПО поддержки – по ссылке [3].

Для грубой оценки количества эффективных разрядов АЦП, то есть таких, которые кодируют собственно входной сигнал, а не внутренние шумы АЦП, соединяем накоротко вход внешней аналоговой цепи выбранного канала АЦП № 7 (контакт J4-7) с аналоговым общим проводом, затем задаём канал 7 в качестве активного и запускаем преобразования. Снятая через параллельный интерфейс с помощью логического анализатора последовательность результатов АЦП для входных выборок 1 MSPS на интервале времени 50 мкс показана на рисунке 30. Как можно видеть из рисунка, младшие три разряда выходного кода АЦП при нулевом напряжении на аналоговом входе дают хаотические всплески единиц, не привязанные по времени ни к каким периодическим сигналам, кроме внутреннего синхросигнала модуля АЦП (1 МГц). Эти всплески отражают влияние на результат преобразования внутренних шумов АЦП. При напряжении полной шкалы 3300 мВ цена младшего значащего разряда АЦП (MЗР, разряд 0) составляет 0,805 мВ, разряда 1 – 1,61 мВ, разряда 2 – 3,22 мВ. Следовательно, максимальная величина всплесков в пересчёте на эквивалентный уровень напряжения при одновременно единичных трёх младших разрядах может составлять до 5,635 мВ.

Аналогичная оценка для канала 0 (выделенный вход АЦП) даёт результат, показанный на рисунке 31. Единичные всплески в разряде 2 здесь происходят реже, чем в предыдущем случае, но также имеют место. Очевидно, в канале 7 шума во входной сигнал добавляет буферный ОУ, который в плане шумовых свойств вовсе не является прецизионным.Таким образом, как для выделенного, так и для универсального входов АЦП по результатам грубой оценки имеем только девять эффективных разрядов. Шумовые свойства АЦП ПЛИС семейства MAX10 на практике оставляют желать лучшего.

Таблица 11. Результаты оценки ошибок смещения нуля и усиления АЦП

Характеристика входного напряжения	Показания ЖКИ макета, мВ	Показания контрольного вольтметра, мВ	Модуль разности показаний, мВ	Доля разности показаний от полной шкалы, %	Примечание
Канал 0 АЦП (выделенный вход АЦП в ПЛИС)					
Нулевое напряжение – замыкание входа АЦП на общий провод	1	–0,2	1,2	0,037	Ошибка смещения нуля
Точка в районе середины шкалы	1703	1700,7	2,3	0,069	Ошибка усиления в середине шкалы
Точка в районе максимума шкалы	3258	3259,3	1,3	0,039	Ошибка усиления у максимума шкалы

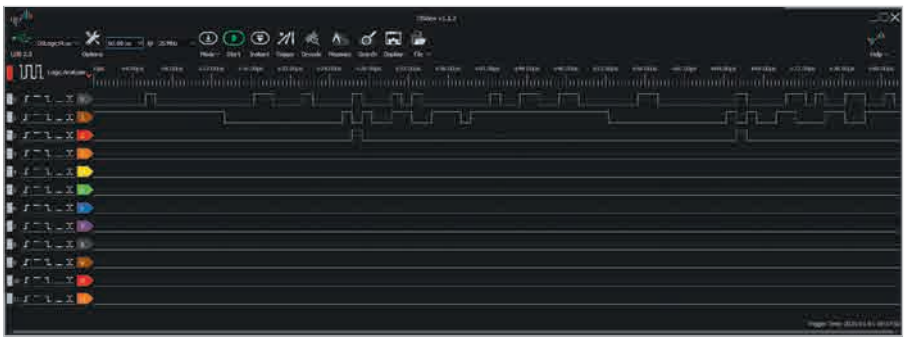

Рис. 31. Результаты АЦП для выделенного входа ПЛИС на интервале 50 мкс для нулевого входного сигнала

Для оценки ошибок смещения нуля и усиления АЦП были произведены измерения напряжения на входе внешней аналоговой цепи выбранного канала 0 АЦП (вывод 3 ПЛИС U2) в нескольких точках в пределах полной шкалы (0...+3,3 В) с помощью контрольного эталонного вольтметра-мультиметра производства RIGOL модели DM3058E, имеющего разрешение $5\frac{1}{2}$ разряда на пределе измерения напряжения 20 В постоянного тока (паспортная погрешность измерений не более 0,02%). Полученные результаты измерений вместе с результатами АЦП, взятыми из показаний ЖКИ макета, приведены в таблице 11. Как можно видеть из таблицы 11, ошибки смещения нуля и усиления в середине и в районе максимума шкалы вполне укладываются в пределы, указанные в таблице 2 (часть 1). Однако из-за внутренних шумов АЦП индицируемые на ЖКИ результаты АЦП постоянно «прыгают» в пределах нескольких милливольт, что затрудняет корректную оценку ошибок смещения нуля и усиления.

По результатам проведённых оценочных исследований можно сделать вывод, что параметры АЦП в ПЛИС семейства MAX10 являются довольно средними как в плане точности, так и в плане скорости. Модуль АЦП в ПЛИС семейства MAX10 по своим характеристикам примерно соответствует 12-разрядными АЦП в большинстве микроконтроллерных семейств. Применение АЦП в ПЛИС MAX10 можно рекомендовать для приложений, не требующих большой точности и скорости: контроль питающих напряжений, оцифровка узкополосных сигналов с непрецизионных датчиков, оцифровка звука со средним качеством и т.п. Для высокоскоростных и высокоточных приложений АЦП в ПЛИС MAX10 рекомендован быть не может.

Оценочное исследование измерителя температуры, выполненного на основе встроенного температурного датчика TSD, показало, что при положительной температуре данный измеритель даёт ошибку порядка 5...7°C, что, впрочем, вполне укладывается в пределы, указанные в таблице 2 (часть 1).

Литература

1. Intel® MAX® 10 Analog to Digital Converter User Guide. Updated for Intel® Quartus® Prime Design Suite: 19.1/ UG-M10ADC | 2020.03.17. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/max-10/ug_m10_adc.pdf.
2. DSLogic Plus USB-based Logic Analyzer. https://www.dreamsourcelab.com/doc/DSLogic_Plus_Datasheet.pdf.
3. https://www.dreamsourcelab.com/download/.