Работа аппаратного USB-моста FTDI FT2232H в режиме синхронного FIFO

Аппаратные USB-мосты компании FTDI являются популярным решением, когда необходимо в короткий срок реализовать USB-интерфейс. Они просты в освоении и сопровождаются драйверами под различные операционные системы. В предыдущей статье [4] были кратко рассмотрены все выпускаемые компанией FTDI аппаратные USB-мосты, а также необходимые программные средства и драйверы для работы с ними. В этой статье мы предлагаем более детально познакомиться с новым режимом работы высокоскоростного USB-моста FT2232H. Это режим «синхронного FIFO», в котором обеспечивается максимальная для аппаратных мостов FTDI скорость передачи данных по USB.

Сергей ДОЛГУШИН
dsa@efo.ru

Микросхема FT2232HL представляет собой двухканальный аппаратный USB-мост, работающий в высокоскоростном режиме USB. Он способен работать в нескольких конфигурациях, основные из которых следующие: UART, асинхронное FIFO, MPSSE и синхронное FIFO. Первые три режима уже известны тем, кто знаком с микросхемами предыдущих серий — FT232, FT245 и FT2232D, и схожи с ними по принципу работы. Отличием аппаратного моста FT2232H от остальных USB-мостов FTDI является новый режим работы — синхронное FIFO. Назначение этого режима — передача данных между компьютером или любым другим USB-хостом и оконечным устройством, если требуется обеспечить скорость обмена более 8 Мбайт/с. По документации производителя, скорость обмена может быть более 25 Мбайт/с. Теоретически пиковые значения скорости передачи могут достигать 52 Мбайт/с, аппаратные возможности микросхемы позволяют обеспечить такую скорость. Для справки: значение 52 Мбайт/с представляет собой максимальную пропускную способность шины USB для режима передачи Bulk. Именно Bulk-режим используется в микросхемах FTDI.

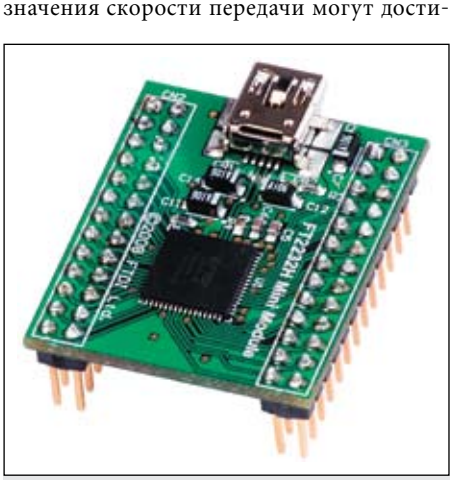

Рис. 1. FT2232HQ Mini Module

Мост FT2232H не является новинкой в данном сегменте, многим разработчикам давно знакомы микросхемы компаний Cypress и NXP (Philips), которые могут быть использованы для решения аналогичных задач. Далее кратко сравним возможности новой микросхемы FTDI и микросхем Cypress.

Микросхемы FTDI, проигрывая по некоторым техническим и функциональным характеристикам, например, микросхемам Cypress серий CY7C6801xA и CY7C68001, существенно выигрывают у них в цене. Основным преимуществом USB-контроллеров Cypress является поддержка всех режимов передачи по USB — Isochronous, Interrupt и Bulk. Это позволяет использовать их в любых USB-приложениях, где требуется гарантированное время доставки (Isochronous) или гарантированная целостность передаваемых данных (Bulk). Мост FT2232H поддерживает передачу только в режиме Bulk.

Следующим пунктом можно выделить энергопотребление. Контроллеры Cypress семейства CY7C6801xA имеют типовое значение потребляемого тока порядка 50 мА (исключая серию CY7C68001, для нее типовое значение составляет 200 мА), для FTDI — порядка 100 мА (значения приводятся на основе информации производителя).

Отметим также более универсальный интерфейс взаимодействия микросхем Cypress с внешним устройством (ПЛИС, ЦСП и т.п.). Сюда относится и возможность выбора разрядности шины данных, она может быть 8-разрядной и 16-разрядной. В целом на пропускную способность это не влияет, но в некоторых случаях упрощает сопряжение с внешним устройством. USB-мост FTDI работает только с 8-разрядной шиной данных. Наличие у микросхем Cypress программируемых порогов наполнения буфера FIFO позволяет более гибко отслеживать состояние последнего. FT2232H дает возможность отслеживать только два статуса: FIFO заполнено и FIFO пустое. Подробнее о работе микросхем CY7C6801xA в режиме Slave FIFO можно прочитать в статье [1].

Дополнительно отметим, что CY7C6801xA имеют встроенное процессорное ядро. Но для режима скоростной передачи данных в режиме Slave FIFO оно не используется, но требует написания программы (firmware). Итогом является усложнение процесса освоения этих контроллеров. Для простых приложений будет предпочтительнее выбрать серию CY7C68001, микросхемы которой не имеют встроенного процессора, и, соответственно, не требуется программирования. Но в последнее время производитель эту серию не развивает. Таким образом, для приложений, где будет достаточно режима Bulk, 8-разрядной шины данных и нет жестких требований по энергопотреблению, аппаратный мост FTDI FT2232H будет разумным выбором.

Оба производителя предоставляют драйверы для работы с их микросхемами. FTDI предлагает их для ОС Windows, MAC и Linux, Cypress — только для Windows (это касается фирменных продуктов).

Аппаратная реализация схожа при использовании микросхем FTDI и Cypress. Схемы

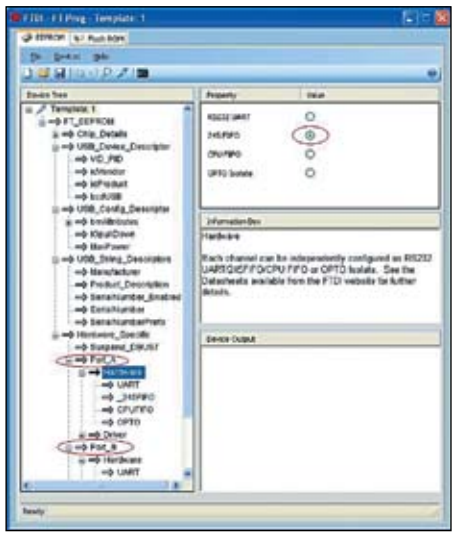

Рис. 2. Выбор режима FT245 FIFO для канала A

Таблица. Список выводов и их функциональное назначение в режиме синхронного FIFO

Номер вывода	Обозначение	Тип	Описание
24,23,22,21, 19,18,17,16	ADBUS[7:0]	Вход/выход	Двунаправленная 8-разрядная шина данных, находится в состоянии «вход», если OE# = 1
26	RXF#	Выход	При логическом 0 указывает на наличие доступных для чтения данных в буфере TX
27	TXE#	Выход	При логическом 0 указывает на свободное место в приемном буфере RX. То есть данные могут быть записаны в буфер RX
28	RD#	Вход	Сигнал чтения. При установке логического 0 разрешает передачу данных из FIFO на линии ADBUS[7:0]. Передача каждого следующего байта происходит по нарастающему фронту тактового сигнала на каждый такт, пока на вход RD# подается логический 0
29	WR#	Вход	Сигнал записи, при установке логического 0 разрешает запись в приемный буфер. Запись каждого следующего байта происходит по нарастающему фронту тактового сигнала на каждый такт, пока на вход WR# подается логический 0
32	CLKOUT	Выход	Тактовый сигнал с частотой 60 МГц. Все управляющие сигналы должны быть синхронизированы с ним
33	OE#	Вход	Сигнал управления направлением передачи по шине данных. Активный уровень — логический 0 — переводит шину в состояние «выход». В режиме чтения он должен быть установлен в активное состояние как минимум за один такт до установки в активное состояние сигнала RD#
30	SIWU	Вход	Сигнал "Send Immediate/WakeUp" имеет двойное назначение: • Если USB находится в режиме Suspend, что индицируется установкой PWREN# = 1, подача короткого импульса с низким активным уровнем приводит к возобновлению обмена по USB. • Если USB находится в обычном режиме (PWREN# = 0), подача короткого импульса с низким активным уровнем приводит к немедленной передаче данных в следующей транзакции, независимо от количества данных в буфере. В обычной ситуации данные будут передаваться только после приема всех 512 байт в буфер

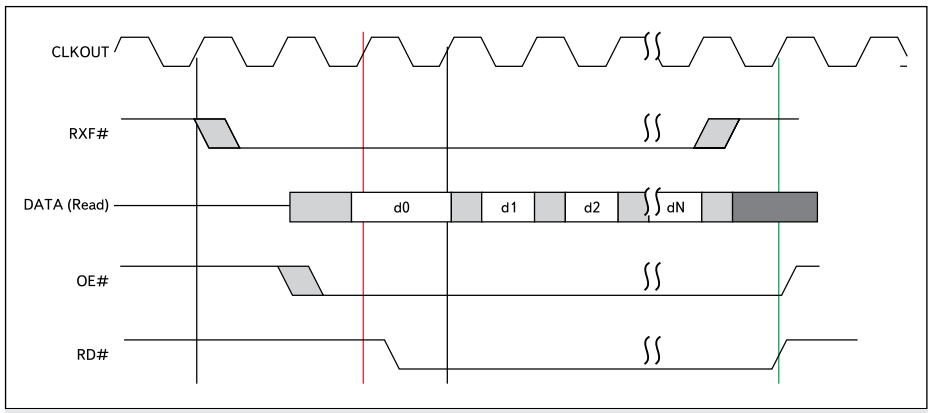

Рис. 3. Режим чтения данных из моста FT2232H

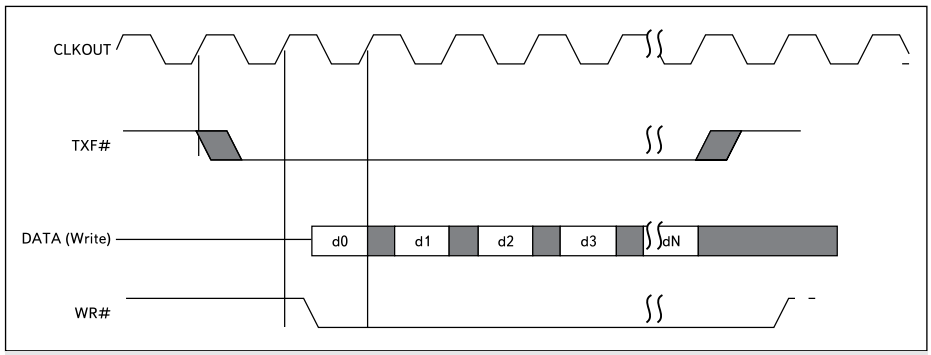

Рис. 4. Режим передачи данных в FT2232H

включения для указанных микросхем требуют установки EEPROM, преобразователя напряжения 5 В –> 3,3 В при использовании питания от шины USB и кварцевого генератора.

Преимуществами рассматриваемого моста FTDI можно назвать цену: она существенно меньше, чем у Cypress, и наличие недорогих отладочных модулей FT2232HQ Mini Module (рис. 1), в отличие от дорогостоящих отладочных наборов Cypress. Причем модули FTDI могут быть использованы не только для ознакомительных и отладочных целей, а также для интеграции в существующие приборы и устройства в качестве мезонинных.

Покажем на базе примера работы с микросхемой FT2232H в режиме синхронного FIFO весь процесс разработки приложений верхнего и нижнего уровней, чтобы дать возможность читателям самостоятельно оценить простоту освоения мостов FTDI.

Работа микросхемы в режиме синхронного FIFO осуществляется под управлением драйвера D2xx в установленном для канала A режиме FT245 FIFO. Скриншот окна программы FTProg показывает указанную настройку режима работы (рис. 2). Далее, с помощью той же программы FTProg параметры новой конфигурации записываются во внешнюю EEPROM, откуда они будут загружены при новом включении моста. Все ресурсы канала B используются совместно с ресурсами канала A и недоступны для приложения; его настройка не влияет на работу канала A. Других специальных аппаратных настроек в общем случае не требуется. Дальнейшая настройка режима синхронного FIFO осуществляется программно, с помощью команды FT_SetBitMode (ftHandle, 0x40). Подробнее данный вопрос рассмотрим в примере программы ниже. Завершение установки режима можно проконтролировать визуально, с помощью осциллографа, наблюдая тактовый сигнал с частотой 60 МГц на выводе 32 (CLKOUT) микросхемы. При использовании синхронного FIFO размеры приемного и передающего буферов составляют 2×512 байт для каждого, а не 4 кбайт, доступные в других конфигурациях. Размер буферов привязан к максимальному размеру пакета в режиме передачи Bulk USB 2.0, равному 512 байт. Этого вполне достаточно для обеспечения максимальной пропускной способности.

В таблице приведены все используемые выводы микросхемы FT2232H и их функциональное назначение, на рис. 3 и 4 приведены диаграммы обмена между мостом и сопрягаемым устройством (ПЛИС, МК и т. п.).

В качестве иллюстрации работы микросхемы FT2232H в режиме синхронного FIFO покажем реализацию тестовой схемы на базе отладочных модулей FT2232HQ Mini Module и Morph IC (рис. 5). Как комментарий к приведенному примеру необходимо сказать, что используемая отладочная плата Morph IC в настоящее время не производится. В качестве функционального аналога можно использовать модуль DLP-FPGA, реализованный на базе ПЛИС Xilinx (рис. 6). Также в ближайшие планы FTDI входит запуск в производство нового модуля на базе ПЛИС Altera — Morph-IC-II. На модуле будет установ-

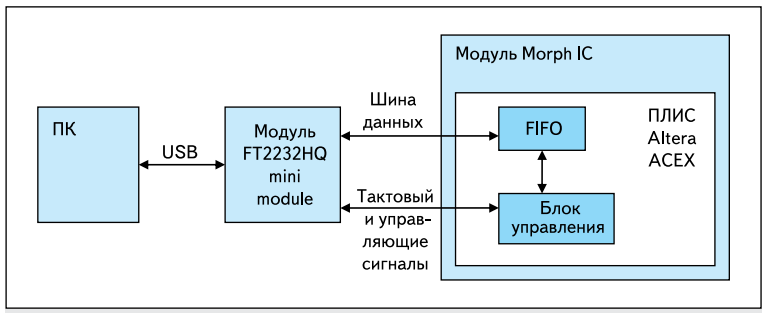

Рис. 5. Тестовая блок-схема

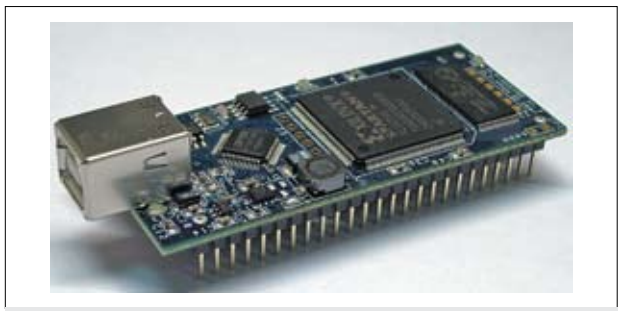

Рис. 6. Отладочный модуль DLP-FPGA

лена ПЛИС EP2C5F256C8N семейства Cyclone 2 и мост FT2232H.

На базе ПЛИС Altera ACEX реализована схема, состоящая из блока управления и FIFO. Блок управления контролирует состояние входных и выходных буферов аппаратного моста FT2232H и формирует сигналы чтения RD, OE и записи WR. Также он осуществляет управление блоком FIFO, реализованным в ПЛИС. Данное приложение выполняет запись данных из ПК в FIFO ПЛИС и передачу их обратно при заполнении данного FIFO, размер которого выбран равным 1024 байт. Рассматриваемый пример показывает все необходимые операции для передачи и приема данных мостом FT2232H в режиме синхронного FIFO.

Программа для ПК была написана в среде Microsoft Visual Studio 6.0, для ПЛИС — в САПР Quartus II 9.0 sp 2. В статье приведены только основные фрагменты проектов, которые показывают все необходимые аспекты работы. Исходные проекты приведенных примеров можно получить по запросу у автора статьи.

Программа верхнего уровня включает три основных блока: инициализация и настройка моста, передача данных и прием данных.

Блок инициализации и настройки:

```
// Определение подключенных устройств FTDI, выбор требуемого (в данном примере — канал A модуля FT2232H Mini Module) и установление с ним обмена

ftStatus=FT_ListDevices(&numDevs,NULL,FT_LIST_NUMBER_ONLY);
    if(ftStatus==FT_OK)
        {
        ftStatus = FT_OpenEx("FT2232H MiniModule A",FT_OPEN_BY_DESCRIPTION,&ftHandle); // данная функция открывает одно из подключенных к ПК устройств FTDI по его названию, в данном случае канал A микросхемы FT2232H,
        }
    else
        {
         // обработка ошибок
        }

//Установка выбранного режима работы моста, следующая последовательность команд настраивает мост для работы в режиме синхронного FIFO

    ftStatus = FT_SetBitMode(ftHandle, 0xff, 0x00); // сброс настроек перед установкой выбранного режима работы
    Sleep(10);

    ftStatus = FT_SetBitMode(ftHandle, 0xff, 0x40); //команда выбора режима работы моста, значение 0x40 соответствует режиму Sync FIFO
    if (ftStatus == FT_OK)
        {
            ftStatus = FT_SetUSBParameters(ftHandle,1024,1024); // выбор размера USB-буферов, по умолчанию равен 4 кбайт, в высокоскоростном режиме передачи рекомендуется устанавливать размер 64 кбайт.
ftStatus = FT_SetFlowControl(ftHandle,FT_FLOW_RTS_CTS,0x10,0x13); // Использование данной команды в приведенном формате предотвращает возможную потерю данных.
        }
    else
        {
        // обработка ошибок
        }
```

При завершении работы данного блока мост FT2232HQ переводится в режим синхронного FIFO, что можно проконтролировать по появлению тактового сигнала CLKOUT частотой 60 МГц на выводе разъема CN2-24 модуля FT2232H Mini Module или выводе 32 микросхемы.

Следующий фрагмент кода показывает блоки программы, в которых выполняется чтение данных из устройства и передача данных:

```
// Прием данных

    FT_GetStatus(ftHandle,&RxBytes,&TxBytes,&EventDWord);
    // проверяем наличие данных в приемном буфере микросхемы
        if (RxBytes > 0)
        {
        ftStatus = FT_Read(ftHandle,RxBuffer,RxBytes, &BytesReceived); // по этой команде считываем данные из микросхемы, RxBuffer содержит сами данные, RxBytes — количество доступных для чтения данных, BytesReceived — число принятых данных
                if (ftStatus == FT_OK)
                    {
                    ptr = RxBuffer;
                    for (int x = 0; x<(int)RxBytes; x++)
                        {
                            st2.Format(" %02X ",*ptr++);
                        st += st2;
                    }

// запись в файл строки st
// создание стандартной панели выбора файла SaveAs
CFileDialog DlgSaveAs(FALSE,(LPCSTR)"txt",NULL, OFN_HIDEREADONLY|OFN_OVERWRITEPROMPT, (LPCSTR)" Text Files (*.txt) |*.txtll");
// отображение стандартной панели выбора файла SaveAs
if(DlgSaveAs.DoModal()==IDOK)
    {
    // создание объекта и открытие файла для записи
    CStdioFile File(DlgSaveAs.GetPathName(), CFile::modeCreate|CFile::modeWrite|CFile::typeBinary);
    // запись в файл строки
    File.WriteString((LPCTSTR)st);
    }
}
else
{
// обработка ошибок
}
}

// Передача данных

    for (i = 0; i <= 1023; i++)
        {
            TxBuffer[i] = x; // запись в буфер данных для дальнейшей передачи,
            x — переменная типа Char
        }
    ftStatus = FT_GetStatus(ftHandle,&RxBytes,&TxBytes, &EventDWord); // проверка готовности устройства к приему данных
```

Рис. 7. Отладочный модуль Morph-IC-II

```
    if((ftStatus == FT_OK) && (TxBytes == 0))
        {
            ftStatus = FT_Write(ftHandle, TxBuffer, sizeof(TxBuffer), &BytesWritten);
                    // передаем данные в устройство
        if (ftStatus == FT_OK)
            {
                    // Завершение функции без ошибки
            }
        else
            {
                    // Обработка ошибок
            }
        }
```

Далее приведем исходный код для блока управления, реализованного на ПЛИС. Его функциональным назначением является формирование сигналов управления OE, RD и WR в соответствии с диаграммами, приведенными на рис. 3 и 4. Также этот блок формирует все внутренние сигналы управления для работы FIFO и двунаправленного порта ввода/вывода. Вопросы реализации блока FIFO и двунаправленного порта ввода/вывода ПЛИС не представляют большого интереса с точки зрения организации обмена между ПЛИС и FT2232H. Кроме того, их реализация и логика работы могут различаться в зависимости от выбранной элементной базы. Основная цель данного примера — показать один из возможных вариантов реализации требуемой диаграммы обмена, что выполняет приведенная ниже программа:

```
entity control is
    port
    (
        RXF : in std_logic; — сигнал наличия данных в буфере FT2232H (см. таблицу 1)
        CLK : in std_logic; -- тактовый сигнал 60 МГц, к нему должны быть привязаны сигналы OE, RD и WR (см. таблицу 1)
        RST : in std_logic; — сигнал сброса, формируемый при включении
        TXE : in std_logic; — сигнал FT2232H о готовности к приему данных (см. таблицу 1)
```

```
      FIFO_F : in std_logic; — FIFO ПЛИС заполнено
      FIFO_E : in std_logic; — FIFO ПЛИС пустое
      OE   : out std_logic; — сигнал разрешения (см. таблицу 1)
      WR   : out std_logic; — (см. таблицу 1)
      RD   : out std_logic; — (см. таблицу 1)
      IN_OUT : out std_logic  — выбор направления передачи порта ПЛИС
   );
end entity;

architecture rtl of control is

-- вспомогательные внутренние сигналы
   signal RD_CONTROL: sr_length;
   signal direction : std_logic;
   signal wr_control : std_logic;

begin

READ_STR: process (CLK, RXF, RST, FIFO_F) — формирование сигналов OE и RD, сигнал RD должен устанавливаться в активный уровень не ранее, чем через один такт после установки в активный уровень сигнала OE. Одновременно эти сигналы в активный уровень устанавливать нельзя.
   begin
      if (RST = '0' and FIFO_F = '1') then
         RD_CONTROL <= "11"; — чтение завершается при заполнении FIFO ПЛИС или наличии сигнала RST
      elsif ((clk'EVENT) and (clk='1')) then
         if (RXF = '0' and wr_control = '1') then — формирование сигналов OE и RD разрешено при наличии данных в передающем буфере FT2232H и завершении цикла передачи из ПЛИС в FT2232H
            RD_CONTROL(1)<= '0';
               RD_CONTROL(0) <= RD_CONTROL(1);
            else
            RD_CONTROL <= "11";
            end if;
      end if;

   OE <= RD_CONTROL(1);
   RD <= RD_CONTROL(0);

end process READ_STR;

WRITE_STR: process (CLK, TXE, RST, FIFO_E) — формирование сигнала WR
   begin
      if (RST = '0' OR RD_CONTROL <= "00" OR FIFO_E = '1') then
         wr_control <= '1'; — передача запрещена, пока не закончится цикл чтения, не заполнится FIFO или в случае сброса

      elsif ((clk'EVENT) and (clk='1')) then
         if (TXE = '0' and RD_CONTROL <= "11") then
            wr_control <= '0'; — формирование сигнала записи WR разрешено при наличии свободного места в приемном буфере FT2232H и завершении цикла приема из ПЛИС в FT2232H
         else
            wr_control <= '1';
         end if;
      end if;
WR <= wr_control;

end process WRITE_STR;

SET_DIR: process (RD_CONTROL(1)) — выбор направления передачи шины данных ПЛИС
   begin
      if (RD_CONTROL(1) = '0') then
         direction <= '0'; — установить шину данных на прием
      else
         direction <= '1'; — установить шину данных на передачу
      end if;
IN_OUT <= direction;
end process SET_DIR;
end rtl;
```

Новый высокоскоростной USB-мост FT2232H расширил возможности применения микросхем FTDI. Благодаря новому режиму работы — синхронному FIFO, а также своей цене и простоте в освоении, он может конкурировать в ряде приложений с микросхемами других производителей.

Литература

1. Долгушин С. Высокоскоростные контроллеры USB производства компании Cypress // Компоненты и технологии. 2006. № 6.
2. Software Application Development D2XX Programmer's Guide.
3. Application Note AN-130 FT2232H Used In An FT245 Style Synchronous FIFO Mode.
4. Долгушин С. Аппаратные USB-мосты FTDI // Компоненты и технологии. 2010. № 4.