6. UART Core

6. Ядро UART

Общее представление о ядре

Ядро UART с интерфейсом Avalon® реализует метод коммуникации потока последовательных символов между встраиваемой системой в Altera® FPGA и внешним устройством. Ядро реализует временные параметры протокола RS-232 и предлагает регулируемую скорость передачи, паритет, стоп бит и биты данных, а также дополнительные контрольные сигналы RTS/CTS. Функциональный набор конфигурируемый, позволяя разработчикам реализовывать только нужную функциональность для выбранной системы.

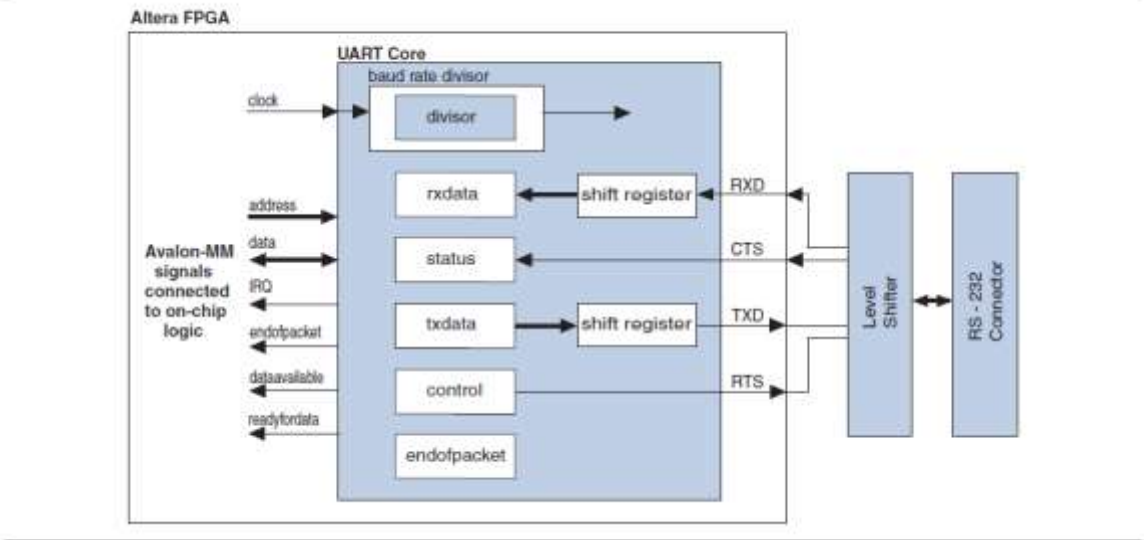
Ядро подключается по слейв интерфейсу Avalon с распределением в памяти (Avalon-MM), который позволяет мастер периферии (такой как процессор Nios II) связываться с ядром посредством простого чтения и записи в регистры контроля и данных. Ядро UART предназначено для SOPC Builder и легко интегрируется в генерируемую систему SOPC Builder. Эта глава состоит из следующих секций:

- "Функциональное описание"
- "Поддержка чипов" на стр. 6-3
- "Инсталляция ядра в SOPC Builder" на стр. 6-3
- "Ограничения при симуляции" на стр. 6-7
- "Программная модель" на стр. 6-8

Функциональное описание

На рис. 6-1 приведена блок-схема ядра UART.

Figure 6–1. Block Diagram of the UART Core in a Typical System



Ядро имеет две части, видимые для пользователя:

- Регистровый файл, с помощью которого осуществляется доступ через слейв порт Avalon-MM
- Сигналы RS-232: RXD, TXD, CTS и RTS

Слейв интерфейс Avalon-MM и регистры

Ядро UART имеет слейв интерфейс Avalon-MM для доступа к внутреннему регистровому файлу. Пользовательский интерфейс с ядром UART состоит из шести 16-битных регистров: control, status, rxdata, txdata, divisor и endofpacket. Мастер периферия, такая как процессор Nios II, использует регистры для контроля над ядром и передачи данных по последовательному интерфейсу.

Ядро UART имеет выход запроса прерывания (IRQ) активной единицей, который может запросить прерывание, когда приняты новые данные, или когда ядро готово к передачи других символов. За подробной информацией обратитесь к секции "Поведение в прерывании" на стр. 6-15.

Слейв порт Avalon-MM способен к передаче с контролем процесса. Ядро UART может быть использовано совместно с периферией прямого доступа к памяти (DMA) с Avalon-MM контролем процесса для автоматизации непрерывного трансферта данных между, например, ядром UART и памятью.

За дополнительной информацией, обратитесь к главе "Ядро интервального таймера" на стр. 27-1. За информацией об интерфейсах Avalon-MM, обратитесь к "Спецификации на интерфейсы Avalon".

Интерфейс RS-232

Ядро UART реализует асинхронную логику приёма и передачи по RS-232. Ядро UART посылает и принимает последовательные данные через порты TXD и RXD. Буферы I/O большинства семейств Altera FPGA не совместимы с уровнями напряжений RS-232, и могут быть повреждены при прямом подключении сигналов к разъёму RS-232. Для совместимости с уровнями напряжений спецификации RS-232 необходим внешний буфер выравнивания уровней между I/O выводами FPGA и внешним разъёмом RS-232 (например, Maxim MAX3237).

Ядро UART использует логический 0 для символа и логическую 1 для пропуска. Инвертор внутри FPGA может быть использован для изменения полярности любых сигналов RS-232, если потребуется.

Логика передатчика

Передатчик UART состоит из 7-, 8- или 9-битного регистра удержания txdata и соответственно из 7-, 8- или 9-битного сдвигового передающего регистра. Мастер периферия на Avalon-MM записывает данные в регистр удержания txdata посредством слейв порта Avalon-MM. Передающий сдвиговый регистр автоматически загружается из регистра txdata, если операция передачи последовательным сдвигом не активна. Сдвиговый регистр передачи напрямую идёт на выход TXD. Данные сдвигаются в TXD, начиная с младшего бита (LSB).

Два этих регистра предлагают двойную буферизацию. Мастер периферия может записывать новые данные в регистр txdata, пока предыдущие записанные данные находятся в процессе сдвига. Мастер периферия может контролировать статус передатчика, читая биты регистра status: передатчик готов (TRDY), сдвиговый регистр передатчика пуст (tmt) и ошибка переполнения передатчика (TOE).

Логика передатчика автоматически вставляет корректное число битов start, stop и parity в последовательный поток TXD данных, как этого требует спецификация RS-232.

Логика приёмника

Приёмник UART состоит из 7-, 8- или 9-битного сдвигового регистра приёма и соответственно из 7-, 8- или 9-битного регистра удержания rxdata. Мастер периферия на Avalon-MM читает данные из регистра удержания rxdata посредством слейв порта Avalon-MM. Регистр удержания rxdata загружается автоматически из сдвигового регистра приёмника каждый раз, когда новый символ полностью передан.

Два этих регистра предлагают двойную буферизацию. Регистр rxdata может удерживать предыдущий переданный символ, пока следующий символ находится в процессе сдвига в регистре приёмника.

Мастер периферия может контролировать статус приёмника, читая биты регистра status: чтение готово (RRDY), детектирование сбоя (BRK), ошибка паритета (PE), ошибка переполнения приёмника (ROE) и ошибка кадрирования (FE). Логика приёмника автоматически вставляет корректное число битов start, stop и parity в последовательный поток RXD данных, как этого требует спецификация RS-232. Логика приёмника проверяет четыре исключительных состояния: ошибка кадрирования, ошибка паритета, ошибка переполнения приёмника и сбой – в принимаемых данных, и устанавливает соответствующие биты в регистре status.

Генерирование скорости обмена

Внутренний тактовый сигнал ядра UART получается из тактового сигнала на входе Avalon-MM. Внутренний тактовый сигнал генерируется делителем частоты. Значение делителя может быть одним из следующих:

- постоянное значение, заданное на стадии генерирования системы
- 16-битное значение, хранимое в регистре делителя

Регистр делителя (divisor) – это дополнительное аппаратное средство. Если он запрещён на стадии генерирования системы, то значение делителя фиксированное, а скорость обмена не может быть изменена.

Поддержка чипов

Ядро UART поддерживается всеми семействами чипов Altera®.

Инсталляция ядра в SOPC Builder

Инсталлирование UART аппаратно создаёт не менее двух I/O портов на каждое ядро UART: вход RXD и выход TXD. Дополнительно аппаратно можно сделать сигналы контроля над процессом: вход CTS и выход RTS.

Используйте интерфейс MegaWizard™ для ядра UART в SOPC Builder для конфигурирования набора аппаратных средств. В следующих секция описаны возможные опции.

Настройки конфигурации

В этой секции описываются настройки конфигурации.

Опция Baud Rate

Ядро UART может реализовывать любую из стандартных скоростей обмена для протокола RS-232. Скорость обмена может быть сконфигурирована одним из двух способов:

- **Фиксированная скорость**– скорость зафиксирована на стадии генерирования системы и не может изменяться через слейв порт Avalon-MM.
- **Изменяемая скорость** – скорость может варьироваться, в зависимости от значения в делителе частоты, хранимое в регистре divisor. Мастер периферия изменяет значение скорости обмена, записывая новые значения в регистр divisor.

Скорость обмена высчитывается на основе тактовой частоты интерфейса Avalon-MM. Аппаратное изменение системной тактовой частоты без регенерации аппаратного ядра UART приводит к некорректным сигналам.

Настройка Baud Rate (bps)

Настройка **Baud Rate** определяет скорость обмена по умолчанию после сброса. Опция **Baud Rate** предлагает стандартные предустановленные значения.

Значение скорости обмена используется для вычисления соответствующего значения делителя, чтобы реализовать необходимую скорость обмена. Скорость обмена и значение делителя взаимосвязаны, как это показано в формуле 6-1 и 6-2.

Equation 6–1.

$$\text{divisor} = \text{int}\left(\frac{\text{clock frequency}}{\text{baud rate}} + 0.5\right)$$

Equation 6–2.

$$\text{baud rate} = \frac{\text{clock frequency}}{\text{divisor} + 1}$$

Настройка Baud Rate Can Be Changed By Software

Когда эта настройка включена, аппаратно реализуется 16-битный регистр делителя по адресу офсета 4. Регистр делителя доступен для записи, таким образом, скорость обмена может быть изменена записью нового значения в этот регистр.

Когда эта настройка выключена, регистр делителя не включается в аппаратную реализацию UART. Аппаратная реализация UART будет иметь постоянный делитель частоты, чьё значение не сможет изменяться после генерирования системы. В этом случае запись по адресу офсета 4 не имеет действия, а чтение по адресу офсета 4 приведёт к непредвиденным результатам.

Настройка Data Bits, Stop Bits, Parity

Бит паритета, биты данных и стоп биты в ядре UART являются конфигурируемыми. Эти настройки фиксируются на стадии генерирования системы; они не могут изменяться посредством регистрового файла. В табл. 6-1 описаны эти настройки.

Табл. 6-1. Настройки битов данных

Настройка	Легальное значение	Описание
Data Bits	7, 8, 9	Эта настройка определяет ширину регистров txdata, rxdata и endofpacket.
Stop Bits	1, 2	Эта настройка определяет, будет ли ядро передавать 1 или 2 стоп бита вместе с каждым символом. Ядро всегда заканчивает приём транзакции по первому стоп биту и игнорирует следующие стоп биты, в зависимости от этой настройки.
Parity	None, Even, Odd	Эта настройка определяет, будет ли ядро UART передавать символы с проверкой паритета, и будет ли оно проверять паритет принимаемых символов. Когда паритет установлен в **None**, логика передатчика посылает данные без бита паритета, а логика приёмника подразумевает, что входящие данные тоже не имеют бита паритета. Бит PE в регистре status не реализован, он всегда 0. Когда паритет установлен в **Odd** или **Even**, логика передатчика подсчитывает и вставляет необходимый бит паритета в исходящий поток TXD, а логика приёмника проверяет бит паритета во входящем потоке RXD. Если приёмник находит данные с некорректным паритетом, бит PE в регистре status устанавливается в 1. Когда паритет установлен в **Even**, бит паритета равняется 0, когда символ имеет чётное количество битов, иначе бит паритета – 1. Аналогично, когда паритет установлен в **Odd**, бит паритета 0, если символ имеет нечётное количество битов.

Настройка Synchronizer Stages

Настройка **Synchronizer Stages** позволяет вам задать длину цепочки синхронизации регистра. Эта цепь регистра используется, когда появляется вероятность возникновения метастабильности, а его длина позволяет задать время до возникновения сбоя. Длина цепи регистра, однако, влияет на задержки в ядре.

За информацией о метастабильности в чипах Altera, обратитесь к *AN 42: Метастабильность в чипах Altera*. За дополнительной информацией об анализе метастабильности и синхронизации цепи регистров, обратитесь к главе "Пространственно временная оптимизация" в томе 2 Настольной книги Quartus II.

Опция Flow Control

Когда включена опция **Include CTS/RTS pins and control register bits** ядро UART включает в себя следующие средства:

- cts_n (отрицательная логика CTS) – входной порт
- rts_n (отрицательная логика RTS) – выходной порт
- бит CTS в регистре status
- бит DCTS в регистре status
- бит RTS в регистре control
- бит IDCTS в регистре control

Основываясь на этих аппаратный средствах, мастер периферия Avalon-MM может детектировать CTS и передавать RTS сигналы контроля над процессом. Входной CTS и выходной RST порты прямо связаны с битами в регистрах status и control, и не имеют прямого влияния на любую другую часть ядра. Когда используется контроль над процессом, следите за тем, чтобы программа обмена на стороне хоста была также сконфигурирована для контроля над процессом.

Когда выключена настройка **Include CTS/RTS pins and control register bits**, ядро не имеет вышеупомянутого аппаратного средства, а непрерывная запись в UART может терять данные. Биты control/status CTS, DCTS, IDCTS и RTS не реализованы, они всегда 0.

Опция Streaming Data (DMA) Control

Интерфейс Avalon-MM ядра UART опционально реализует Avalon-MM трансферты с контролем над процессом. Контроль над процессом позволяет мастер периферии Avalon-MM записывать данные только, когда ядро UART готово запросить другой символ, а читать данные только, когда данные ядра доступны. Ядро UART может опционально содержать регистр end-of-packet.

Настройка Include End-of-Packet Register

Когда эта опция включена, ядро UARTсодержит:

- 7-, 8- или 9-битные регистры endofpacket по адресу офсета 5. Ширина данных определяется настройкой **Data Bits**.
- бит EOP в регистре status.
- бит IEOP в регистре control.
- Сигнал endofpacket в интерфейсе Avalon-MM для поддержки трансферта данных с контролем над процессом в и из других мастер периферий системы.

Детектирование End-of-packet (EOP) позволяет ядру UART завершать транзакцию данных с мастером Avalon-MM с контролем над процессом. Детектирование EOP может быть использовано с контроллером DMA, например, для реализации UART, который автоматически записывает принятые символы в память, пока специальный символ не появляется во входящем потоке RXD. Значение конечного символа (EOP) задаётся в регистре endofpacket.

Когда регистр EOP запрещён, ядро UART не имеет ресурсов EOP. Запись в регистр endofpacket не имеет значения, а чтение приводит к непредвиденным результатам.

Настройки симуляции

Когда генерируется логика ядра UART, также создаётся модель симуляции. Модель симуляции содержит средства для упрощённой и ускоренной симуляции системы, которая использует ядро UART. Изменения в настройках симуляции не влияют на аппаратное поведение ядра UART; настройки влияют только на функциональную симуляцию.

За примерами использования следующих настроек для симуляции систем Nios II, обратитесь к *AN 351: Симуляция проектов со встроенным процессором Nios II*.

Настройка Simulated RXD-Input Character Stream

Вы можете ввести поток символов, которые симулируют входные данные для порта RXD после сброса симулируемой системы. Интерфейс MegaWizard™ для ядра UART связывает произвольную строку символов с моделью симуляции UART. После сброса, строка поступает в порт RXD символ за символом, так как будто ядро способно принимать новые данные.

Настройка Prepare Interactive Windows

На стадии генерирования системы, генератор ядра UART может создавать макрос ModelSim, который способствует интеграции с моделью UART во время симуляции. Вы можете включить следующие опции:

- **Create ModelSim alias to open streaming output window** – для создания макроса ModelSim, который открывает окно для отображения всех выходов порта TXD.
- **Create ModelSim alias to open interactive stimulus window** - для создания макроса ModelSim, который открывает окно для приёма стимулов для RXD порта.

Настройка Simulated Transmitter Baud Rate

Скорость обмена RS-232 значительно меньше, чем любой другой процесс в системе, поэтому она редко используется для симулирования программной модели на реальной скорости передачи. Например, на 115200 бит/с требуется тысячи тактовых циклов для трансферта одного символа. Модель симуляции UART имеет возможность запускаться с постоянным значением в делителе – 2, позволяя симулируемому UART трансферт битов на половинной скорости от системного такта, это приблизительно один символ за 20 тактовых циклов. Вы можете выбрать одну из следующих опций для скорости передачи симулированного передатчика:

- **Accelerated (use divisor = 2)**—TXD в симуляции выдаёт один бит за два тактовых цикла.
- **Actual (use true baud divisor)**—TXD передаёт на реальной скорости, определённой в регистре divisor.

Ограничения при симуляции

Средства симуляции создаются для простой симуляции процессорных систем Nios II, когда используется симулятор ModelSim. В документации на процессор приведено примерное использование этих средств. Другое использование также возможно, но требует дополнительных усилий пользователя, чтобы создать собственный процесс симуляции.

Модель симуляции находится в HDL файле верхнего уровня ядра UART; HDL для синтеза и HDL для симуляции находятся в одном файле. Средства симуляции реализуются, используя директивы синтеза translate on и translate off (включение/выключение транслирования), которые делают специальные секции HDL кода видимыми только для инструментов синтеза.

Не редактируйте директивы симуляции, если вы используете рекомендованные Altera процедуры симуляции. Если вы меняете директивы симуляции для своего собственного процесса симуляции, следите за тем, чтобы SOPC Builder перезаписывал существующие файлы на стадии генерирования системы. Будьте бдительны, потому что ваши изменения могут быть не перезаписаны.

За подробной информацией о симуляции ядра UART в процессорных системах Nios II обратитесь к *AN 351: Симуляция проектов со встроенным процессором Nios II*.