

IP ядро
Преобразователь AXI-Stream
в UART и обратно

Оглавление

1. Описание IP ядра.....	3
2. Описание UART интерфейса.....	5
3. Описание AXI-Stream интерфейса	7
4. Описание приемной части IP-ядра	9
5. Описание передающей части IP-ядра	14

1. Описание IP ядра

Ядро состоит из двух частей: RX и TX. Часть RX принимает данные из UART и выдает их в виде AXI-Stream потока. Часть TX принимает данные из AXI-Stream потока и выдает их в виде данных для UART интерфейса. Внешний вид ядра представлен на рис. 1.1, в табл. 1.1 описаны входные и выходные порты.

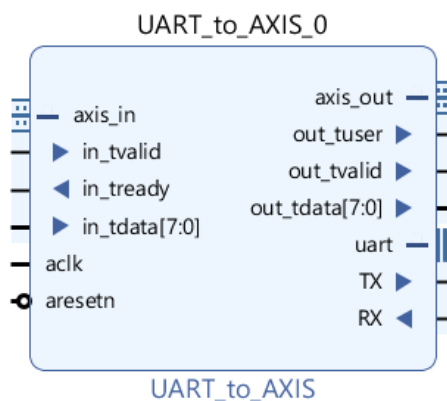


Рис. 1.1 Внешний вид IP ядра

Таблица 1.1 Назначение портов IP ядра

Название	Назначение
aclk	тактовый сигнал
aresetn	синхронный сигнал сброса, активный низкий уровень
in_tdata	данные для передачи в UART, всегда используются младшие биты
in_tvalid	данные для передачи установлены на шине in_tdata
in_tready	ядро готово получить новые данные для передачи в UART
out_tdata	данные принятые по UART, всегда используются младшие биты
out_tuser	флаг ошибки четности; '1' – ошибка четности в принятых данных
out_tvalid	данные на шинах out_tdata и out_tuser установлены
TX	UART TX
RX	UART RX

Ядро является конфигурируемым. На рис. 1.2 представлено окно настроек, в табл. 1.2 представлено описание настраиваемых параметров.

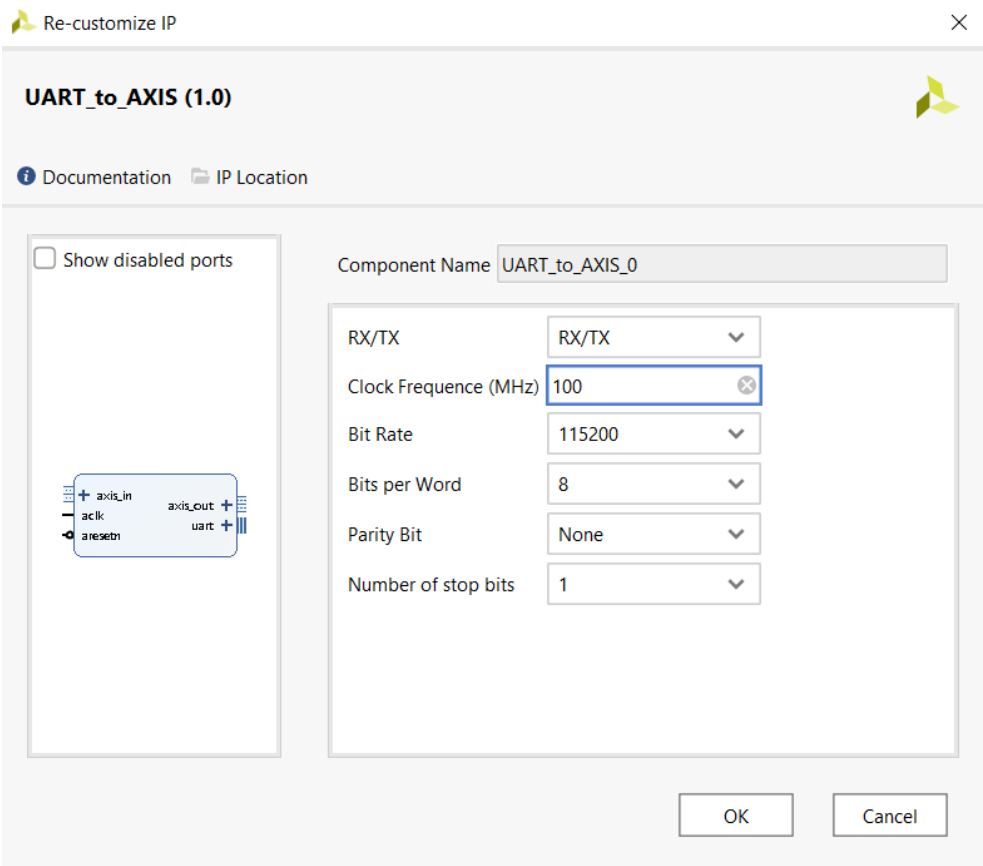


Рис. 1.2 Окно настроек IP ядра

Таблица 1.2 Описание настраиваемых параметров

Название	Назначение
RX/TX	использовать, только RX часть ядра, только TX часть или обе части
Clock Frequency (MHz)	тактовая частота сигнала <code>ack</code> в мегагерцах
Bit Rate	скорость UART интерфейса
Bits per Word	число бит данных в одном слове UART
Parity Bit	настройка бита четности в UART интерфейсе
Number of stop bits	количество стоп-бит в UART интерфейсе

2. Описание UART интерфейса

При отсутствии данных для передачи линия находится в состоянии ожидания (IDLE). Для начала передачи на линию выставляется старт-бит (START). После этого на линию выставляются по одному биты слова данных (DATA), начиная с младшего. Количество бит в слове может быть от 5 до 8.

После этого на линию может выставляться бит четности (PARITY). Бит четности рассчитывается на основе бит данных. Три возможных варианта:

- Even - бит равен '1' при нечетном числе единиц в слове данных;
- Odd - бит равен '0' при нечетном числе единиц в слове данных;
- None - бит четности не используется.

Далее на линию должен выставляться один или два стоп-бита (STOP). После окончания передачи линия переходит в состояние ожидания (IDLE). Следующая передача может начаться сразу после предыдущей, без перехода в состояние IDLE.

Таблица 2.1 Состояние линии и сигналы

Состояние линии	Значение сигналов
IDLE	'1'
START	'0'
DATA0	data[0]
DATA1	data[1]
...	..
DATAN	data[n]
PARITY	рассчитывается
STOP	'1'

Таблица 2.2 Скорости приема и передачи

Возможные скорости передачи (бит/с)	
1200	38400
2400	57600
4800	115200
9600	230400
14400	460800
19200	921600

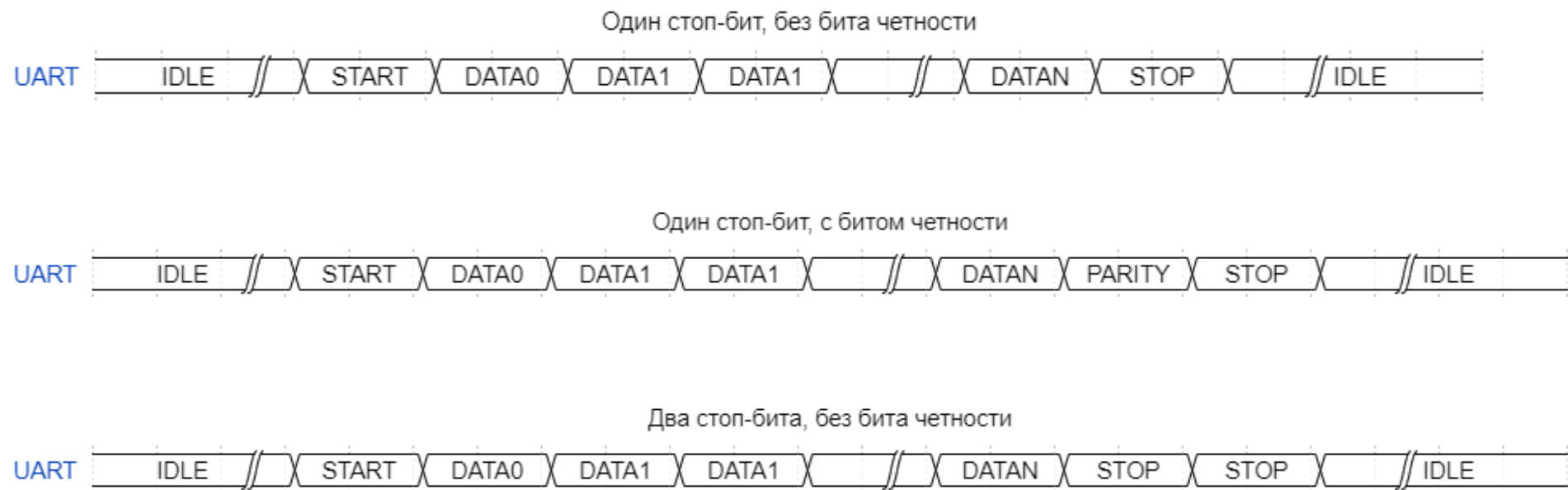


Рисунок 2.1 Возможные временные диаграммы для UART интерфейса

3. Описание AXI-Stream интерфейса

Все изменения сигналов интерфейса осуществляются синхронно с тактовым сигналом ACLK. Сигнал сброса ARESETN является синхронным с активным низким уровнем.

Сигналы TVALID и TREADY с помощью handshake протокола определяют, как будет осуществляться передача данных TDATA и TUSER. Для передачи данных оба сигнал TVALID и TREADY должны быть установлены в '1'. Первым может быть установлен либо TVALID, либо TREADY, либо они оба могут быть установлены на одном и том же такте ACLK.

Master не может ждать установки TREADY перед тем, как установить TVALID. Если TVALID установлен, он должен оставаться установленным до передачи данных по TREADY.

Slave может ждать TVALID перед тем, как установить соответствующий ему TREADY. Если slave установил TREADY, он может его сбросить до установки TVALID.

Таблица 3.1 Назначение сигналов AXI - Stream

Название	Назначение
ACLK	тактовый сигнал
ARESTN	сигнал сброса
TDATA	данные для передачи, 8*N бит
TUSER	дополнительные данные для передачи, К бит
TVALID	данные для передачи доступны на шине
TREADY	приемник готов получить данные



Рисунок 3.1 Возможная временная диаграмма для AXI-Stream интерфейса

4. Описание приемной части IP-ядра

После сброса сигнала **ARESETN** приемник находится в состоянии ожидания данных. По спаду сигнала **RX** приемник устанавливает сигнал **RX_FALLING**, инициализирует и стартует счетчики. Максимальное значение счетчика устанавливается половине числа тактов **ACLK** на период бита (**Cycles per Period - CPP**), чтобы попасть на середину старт-бита.

Досчитав до $CPP/2$, счетчик сбрасывается и его максимальное значение устанавливается равным **CPP**, чтобы попасть на середину бита данных. По окончании счета устанавливается сигнал **CLK_COUNT_DONE**, счетчик сбрасывается и в регистр сдвига записывается первый полученный бит. Счетчик числа принятых бит увеличивается на единицу. Аналогичным образом принимаются остальные биты данных. При приеме последнего бита данных устанавливается сигнал **BIT_COUNT_DONE**.

Если передается бит четности, то он принимается также, как и биты данных. Далее в зависимости от настройки **Even** или **Odd**, определяется приняты данные с ошибкой или нет. Результат сохраняется в регистр ошибок.

Далее принимается стоп-бит, если передается второй стоп-бит он также принимается. После этого выдаем полученные данные на выход и переходим в состояние ожидания новых данных.

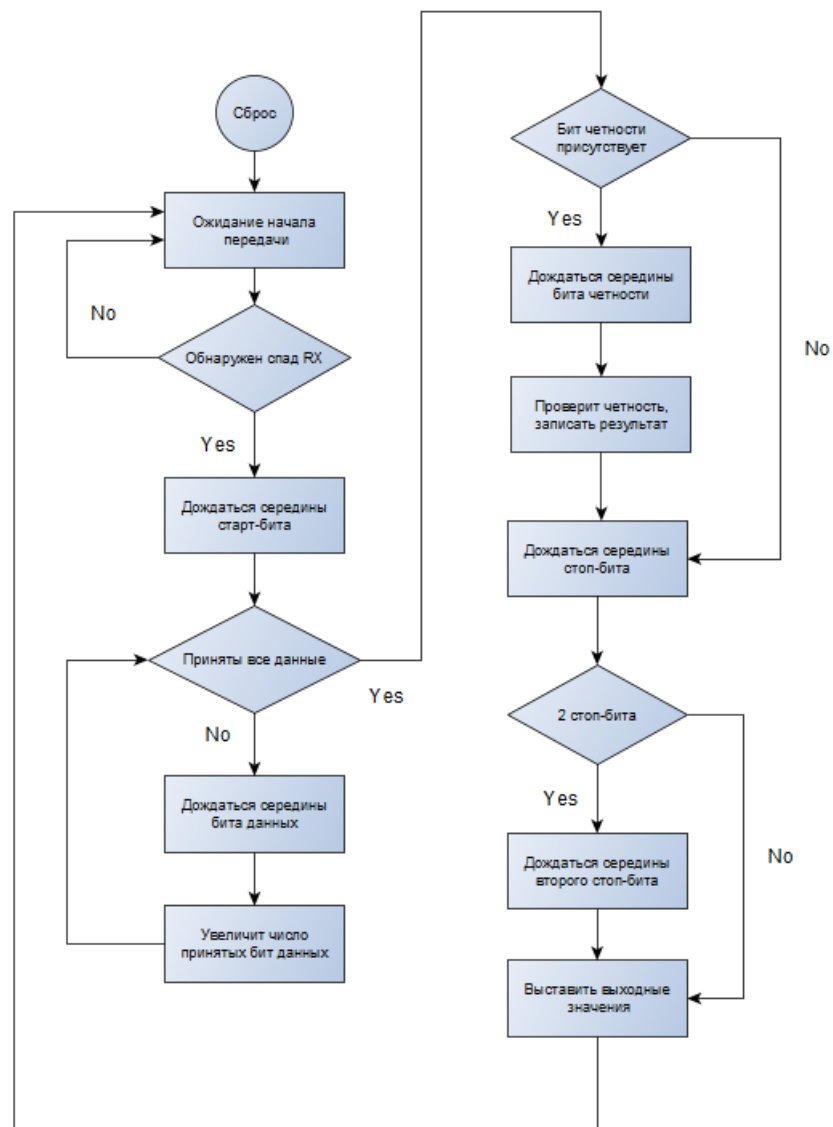


Рисунок 4.1 Блок диаграмма алгоритма приема

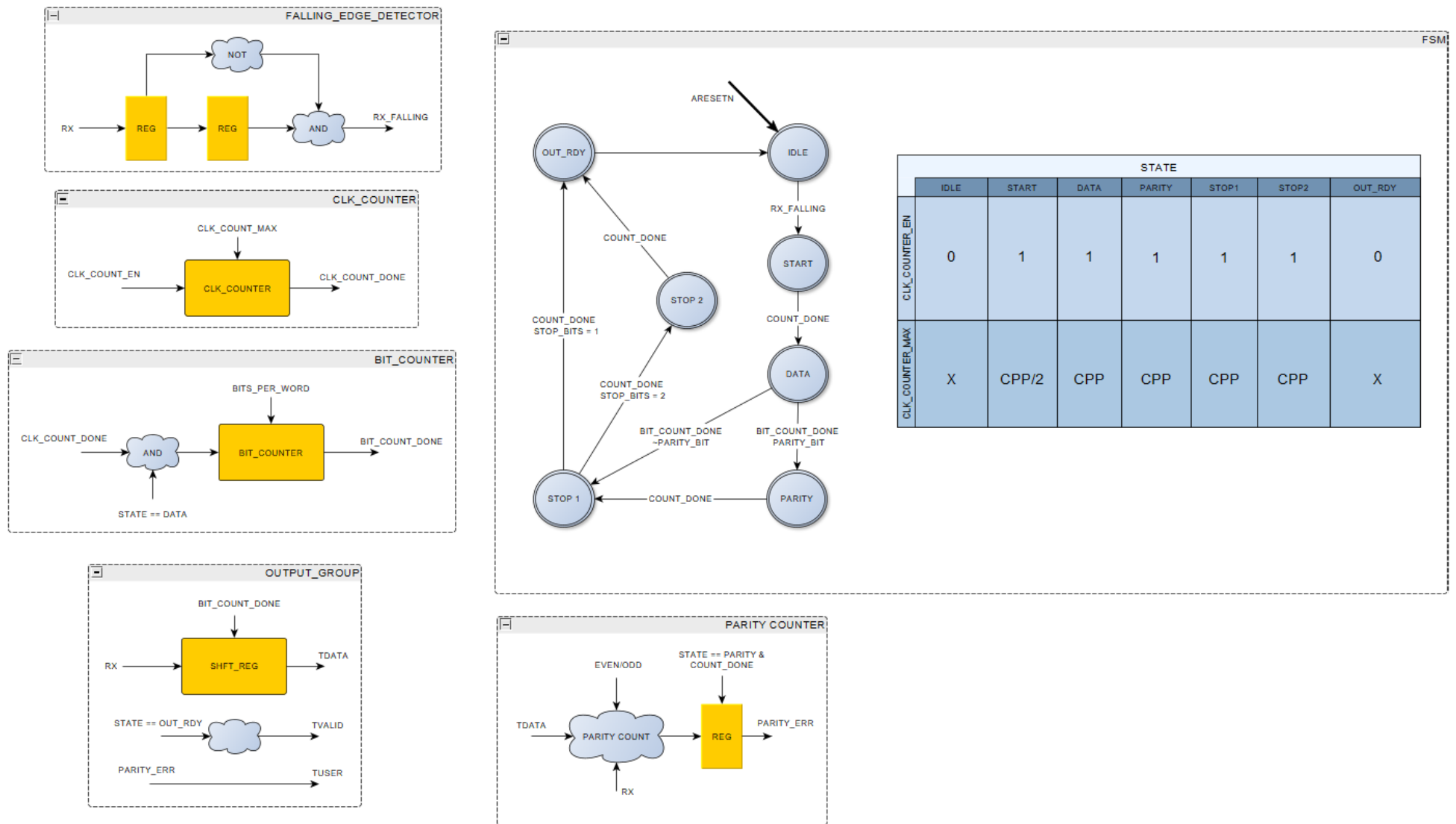
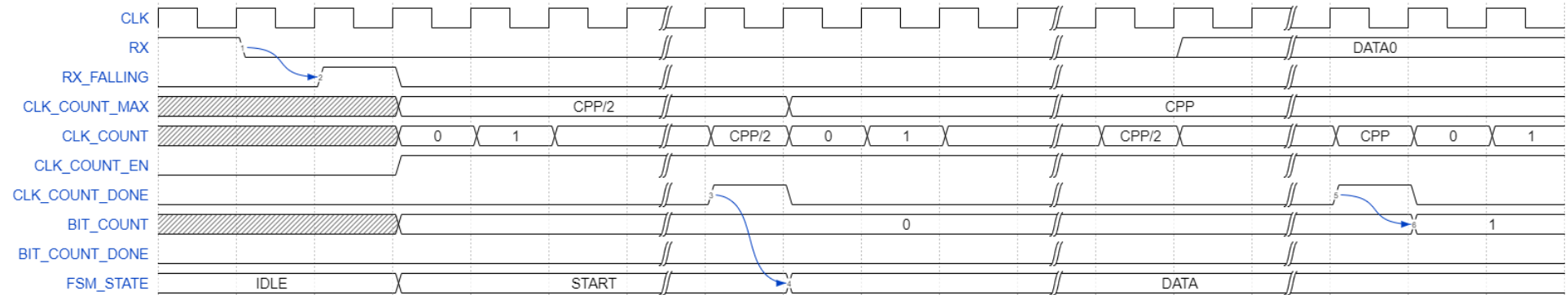
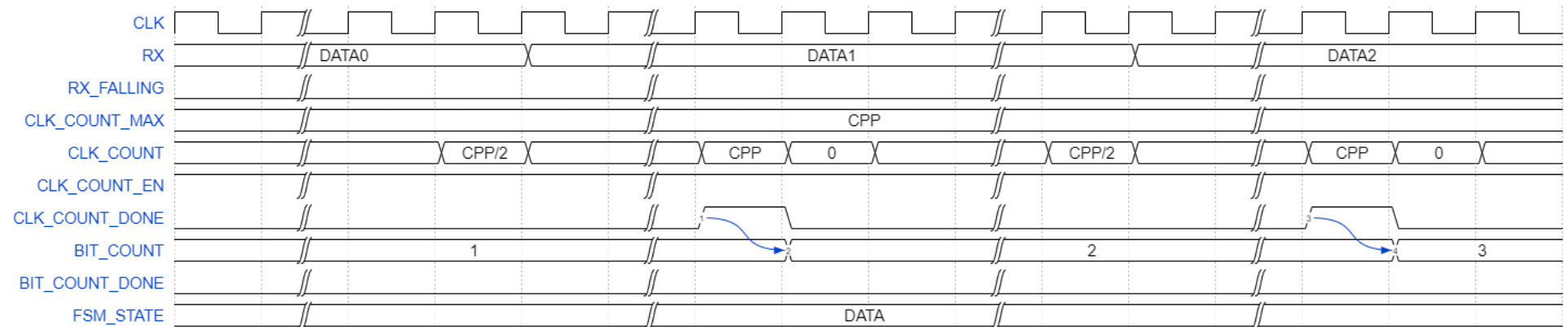


Рисунок 4.2 Блок схема приемной части ядра

Прием старт-бита и первого бита



Прием первых бит данных



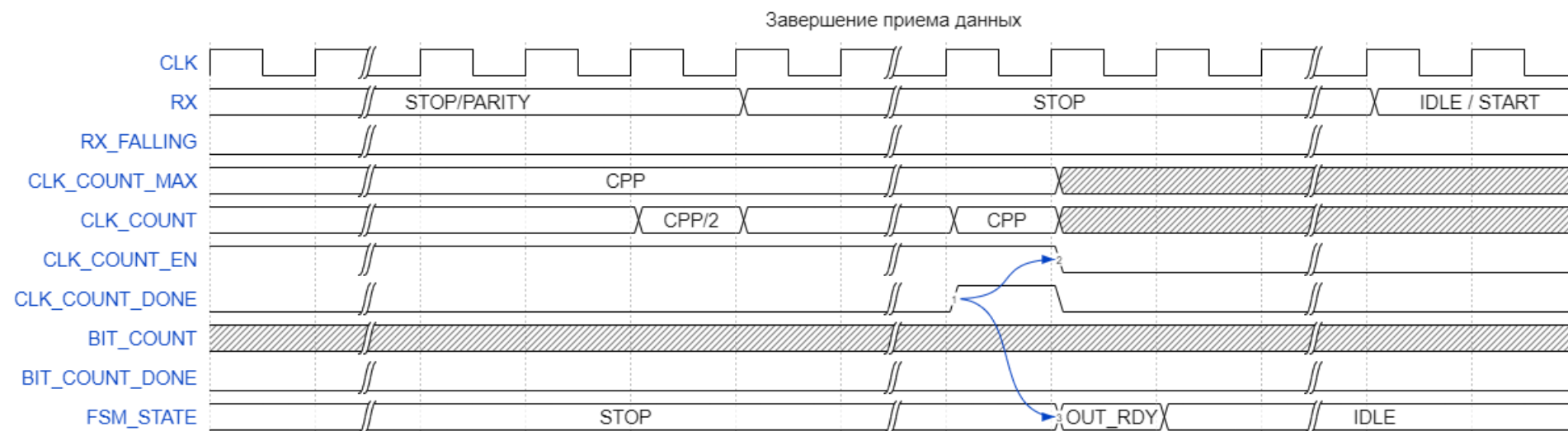


Рисунок 4.3 Временные диаграммы приемной части ядра

5. Описание передающей части IP-ядра

После сброса сигнала **ARESETN** передатчик находится в состоянии ожидания данных и устанавливает **TREADY**, показывая, что он готов принять данные. При установке сигнала **TVALID** передатчик защелкивает входные данные, инициализирует и стартует счетчики и выставляет на выход **TX** старт-бит. Максимальное значение счетчика устанавливается равным числу тактов **ACLK** на период бита (**Cycles per Period - CPP**).

Досчитав до **CPP**, счетчик сбрасывается, устанавливается сигнал **CLK_COUNT_DONE**, на выход **TX** выставляется первый бит данных. Сигнал **CLK_COUNT_DONE** указывает, что бит передан, после чего счетчик числа переданных бит увеличивается на единицу. Аналогичным образом передаются остальные биты данных. После передачи последнего бита данных устанавливается сигнал **BIT_COUNT_DONE**.

Если передается бит четности, то он рассчитывается и выставляется на **TX** также, как и биты данных. Далее выдается один или два стоп-бита. После этого устанавливается сигнал **TREADY** и блок переходит в состояние ожидания новых данных.

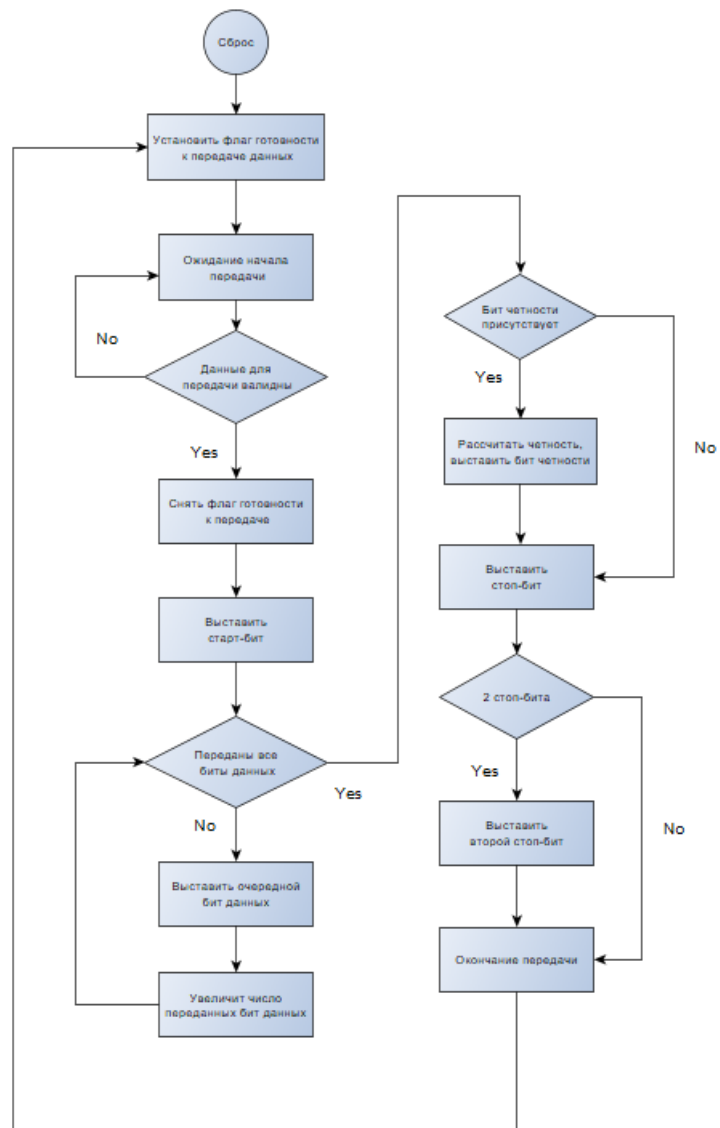


Рисунок 5.1 Блок диаграмма алгоритма передачи

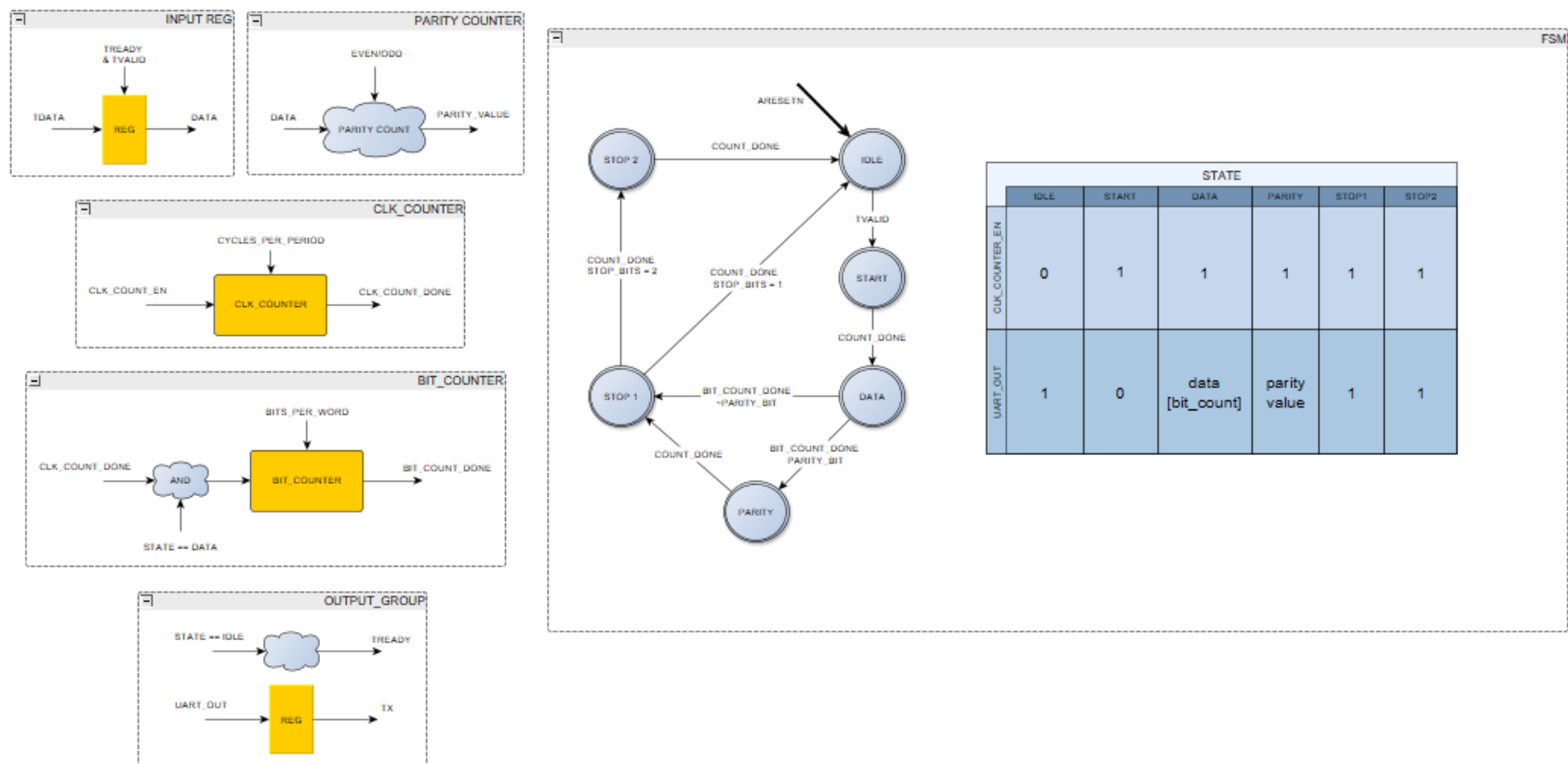
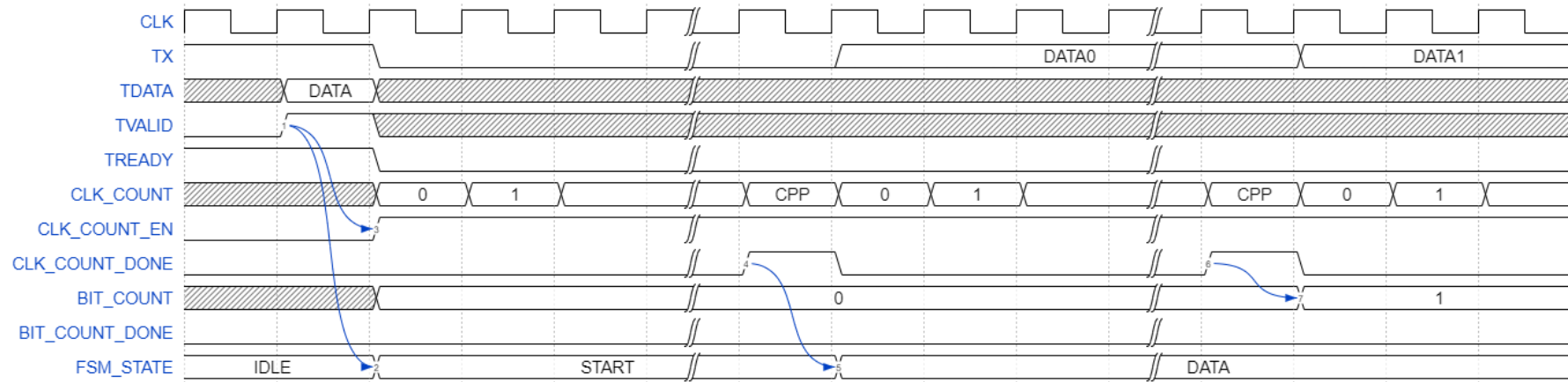
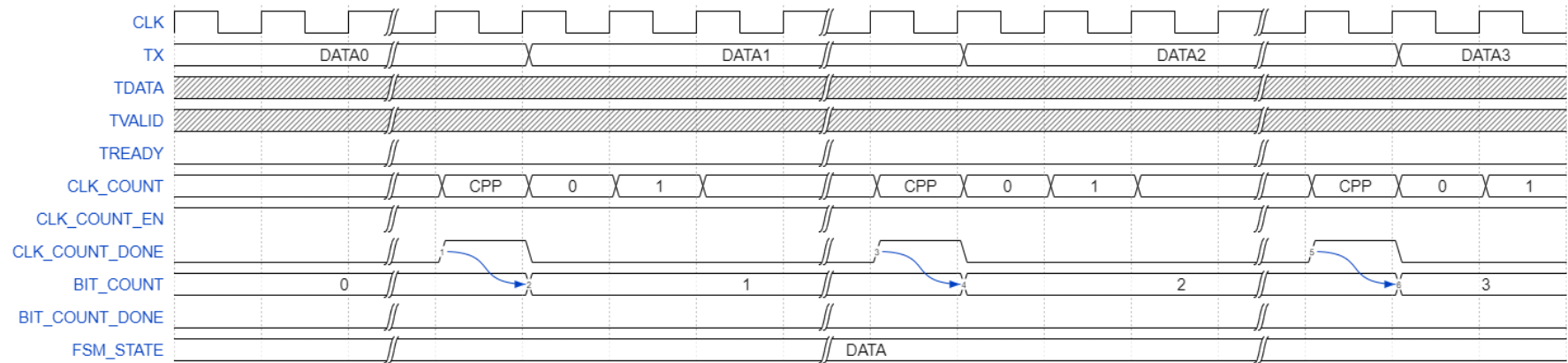


Рисунок 5.2 Блок схема передающей части ядра

Выдача старт-бита и первых бит данных



Выдача первых бит данных



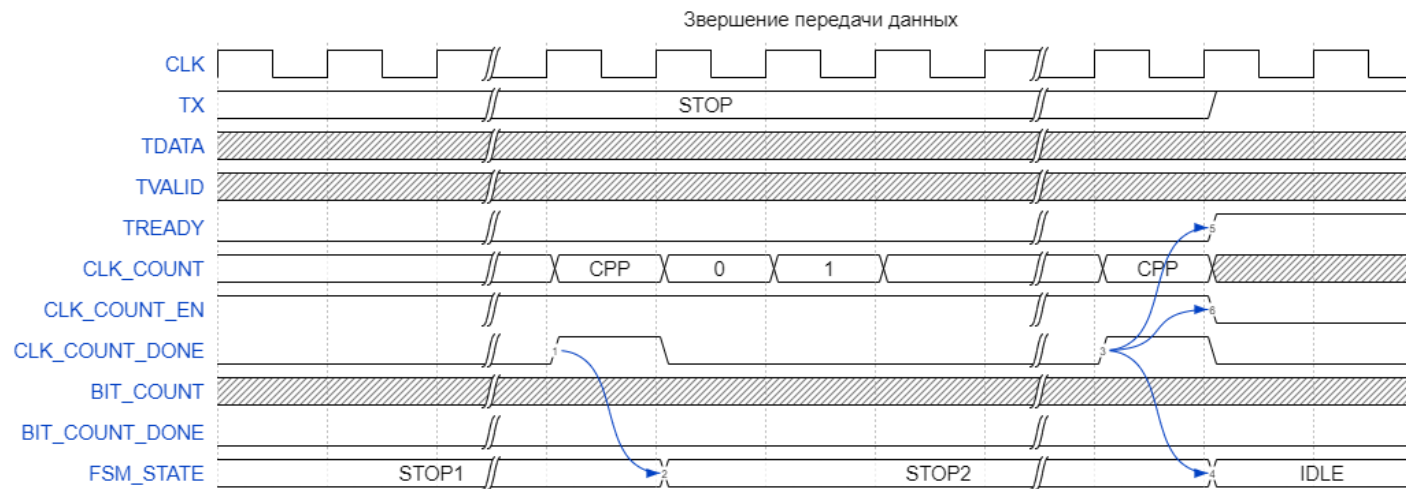
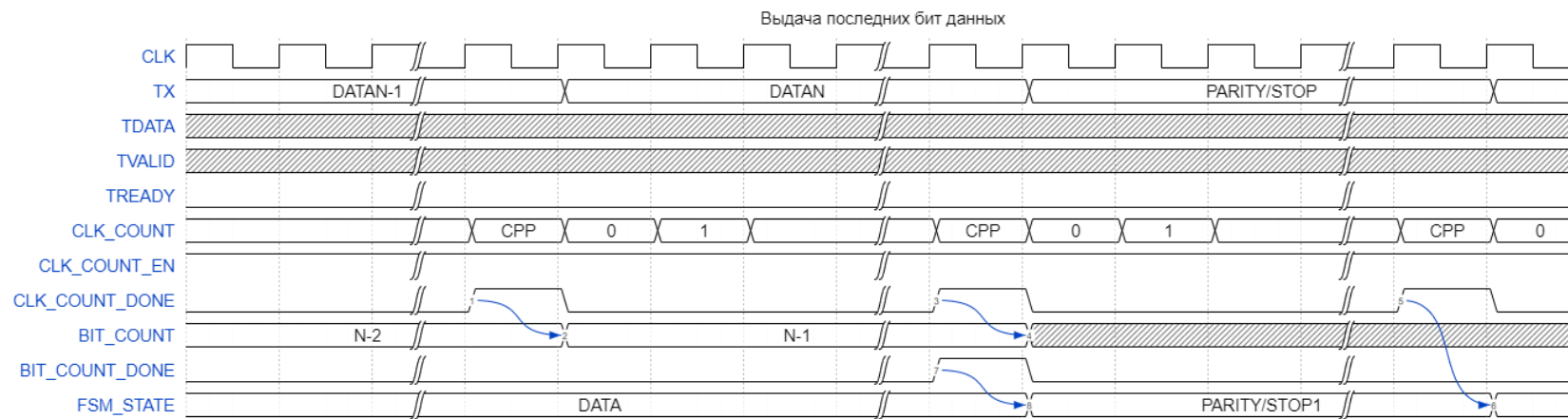


Рисунок 5.3 Временные диаграммы передающей части ядра