

IP ядро

**Взаимная корреляционная функция
комплексных сигналов
на основе быстрой свертки**

Оглавление

1. Описание IP ядра.....	3
2. Общая блок-схема и описание алгоритма.....	6
3. Блок конфигурации FFT IP	9
4. Блок считывания сигнала $f1$	11
5. Блок считывания сигнала $f2$	13
6. Блок считывания BRAM	15
7. Блок комплексного произведения и битового сдвига.....	18
8. Блок отбрасывания нулевых отсчетов.....	19
9. Конечный автомат управления блоками ядра.....	21

1. Описание IP ядра

Ядро выполняет вычисление взаимной корреляционной функции двух комплексных сигналов заданной длины с помощью быстрой свертки. Обозначим сигналы, для которых нужно найти корреляционную функцию, как f_1 и f_2 , а их длины N_1 и N_2 соответственно. Ниже представлены формулы для свертки и корреляции:

$$\begin{aligned} conv[k] &= \sum_{m=0}^{N_1+N_2-1} f_1[m] \cdot f_2^*[k-m]; \\ corr[k] &= \sum_{m=0}^{N_1+N_2-1} f_1[m] \cdot f_2^*[m-k]. \end{aligned}$$

Видно, что формулы отличаются только знаком аргумента у сигнала f_2 , поэтому корреляционную функцию можно рассчитать с помощью свертки сигналов f_1 и f'_2 , где $f'_2[k] = f_2[N_2 - k]$. То есть у сигнала f_2 необходимо изменить порядок следования отсчетов.

Для экономии ресурсов свертку удобно выполнять с помощью алгоритма быстрой свертки. Для этого сигналы f_1 и f'_2 необходимо дополнить нулевыми отсчетами, с помощью FFT найти спектры сигналов, найти произведение спектров и вычислить обратное FFT. После этого из полученного сигнала нужно выделить $N_1 + N_2 - 1$ первых отсчетов, в которых будет храниться корреляционная функция. Размерность FFT следует выбирать исходя из соотношения

$$N_1 + N_2 - 1 < N_{FFT} = 2^L,$$

где N_{FFT} – размерность FFT, L – число стадий FFT. Размерность FFT удобно выбрать константой равной

$$N_{FFT} = 2 \cdot \max\{N_1, N_2\}.$$

Вычисление FFT будет производиться с помощью IP ядра Xilinx. Сначала ядро будет использовано для расчета спектра f_1 , затем для расчета спектра f'_2 и затем для вычисления обратного FFT от произведения спектров.

Внешний вид ядра представлен на рис. 1.1. В табл. 1.1 описаны входные и выходные порты ядра.

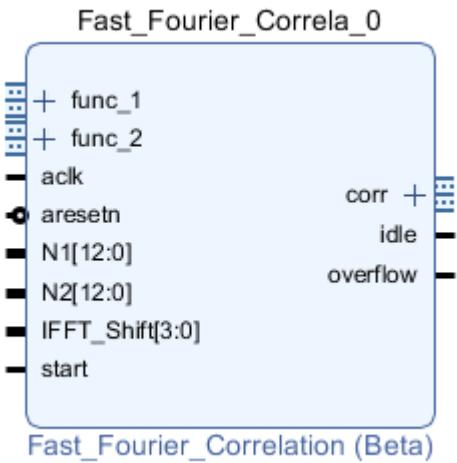


Рис. 1.1 Внешний вид IP ядра

Таблица 1.1 Назначение портов IP ядра

Название	Назначение
aclk	тактовый сигнал
aresetn	синхронный сигнал сброса, активный низкий уровень
N1	число отсчетов в сигнале f_1
N2	число отсчетов в сигнале f_2
IFFT_Shift	битовый сдвиг перед обратным FFT
start	сигнал начала вычисления корреляции
idle	корреляция рассчитана, ядро ожидает start
f1_tdata	отсчеты сигнала f_1 (16 бит I, 16 бит Q)
f1_tvalid	данные установлены на шине f1_tdata
f1_tready	ядро готово получить данные f1_tdata
f2_tdata	отсчеты сигнала f_2 (16 бит I, 16 бит Q)
f2_tvalid	данные установлены на шине f2_tdata
f2_tready	ядро готово получить данные f2_tdata
corr_tdata	данные корреляционной функции
corr_tvalid	данные на шинах corr_tdata установлены
corr_tready	следующее ядро готово принять данные
overflow	флаг переполнения после сдвига

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

Таблица 1.2 Описание настраиваемых параметров

Название	Назначение
NFFT	размерность FFT

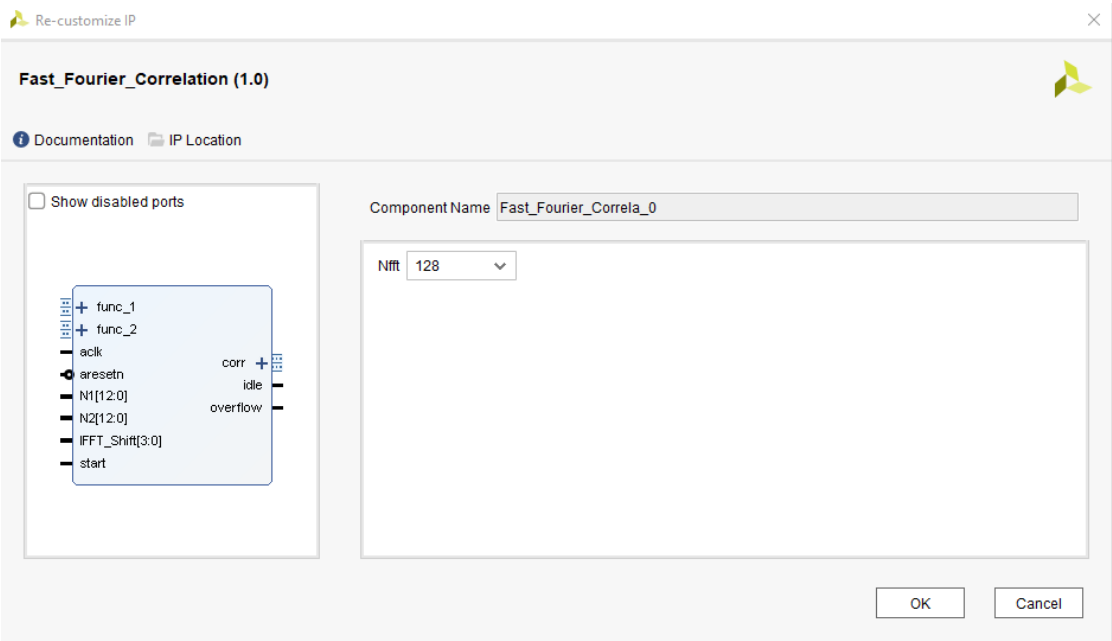


Рис. 1.2 Окно настроек IP ядра

2. Общая блок-схема и описание алгоритма

После сброса ядро находится в состоянии ожидания сигнала начала вычисления и устанавливает сигнал `idle` в '1'. После установки сигнала `start` в '1' сигнал `idle` сбрасывается в '0' и ядро начинает расчет корреляционной функции.

Сначала блок конфигурации ядра Xilinx FFT посылает команду, чтобы рассчитывалось прямое FFT. Далее блок считывания сигнала f_1 получает из FIFO на входе $N1$ отсчетов и передает их на вход мультиплексора. После передачи всех отсчетов блок передает на вход мультиплексора $N_{FFT} - N1$ нулевых отсчетов. С выхода мультиплексора отсчеты поступают на ядро Xilinx FFT, выход которого буферизируется в FIFO 1. Параллельно с этим блок считывания сигнала f_2 получает из FIFO на входе $N2$ отсчетов и записывает их в BRAM в обратном порядке с комплексным сопряжением.

После того как с выхода Xilinx FFT получены все отсчеты и в BRAM записаны все отсчеты блок считывания BRAM получает $N2$ записанных отсчетов и передает их на мультиплексор. После передачи всех отсчетов блок считывания BRAM выдает $N_{FFT} - N2$ нулевых отсчетов. С выхода мультиплексора отсчеты поступают на ядро Xilinx FFT.

Блок вычисления комплексного произведения получает по одному отсчету из FIFO 1 и Xilinx FFT, перемножает их, сдвигает на `IFFT_Shift` бит влево и записывает в FIFO 2. При обнаружении переполнения блок выставляет флаг.

Далее блок конфигурации ядра Xilinx FFT посылает команду, чтобы рассчитывалось обратное FFT. После этого данные из FIFO 2 поступают на вход ядра Xilinx FFT. Отсчеты с выхода Xilinx FFT поступают на блок отбрасывания нулевых отсчетов. Данный блок передает на выход ядра первые $N1 + N2 - 1$ отсчетов. После этого блок переходит в состояние ожидания и устанавливает сигнал `idle` в '1'.

Блок схема ядра представлена на рис. 2.1. Назначение блоков на рисунке описано ниже:

- **FFT IP** – ядро Xilinx FFT (PG 109);
- **FIFO** – ядро Xilinx;
- **BRAM** – ядро Xilinx;
- **Null** – формирует нулевой tvalid и tdata, чтобы остановить FFT IP, когда ядро находится в состоянии ожидания;
- **MUX** – мультиплексор AXI-Stream потоков на входе FFT IP;
- **DEMUX** – демultipлексор потоков на выходе FFT IP;
- **Control FSM** – конечный автомат управления блоками ядра;
- **FFT Config** – блок конфигурации FFT IP;
- **Recv F1** – блок считывания сигнала f_1 ;
- **Recv F2** – блок считывания сигнала f_2 ;
- **Read_BRAM** – блок считывания BRAM;
- **Mult_Shift** – блок комплексного произведения и сдвига;
- **Out_Block** – блок отбрасывания нулевых отсчетов.

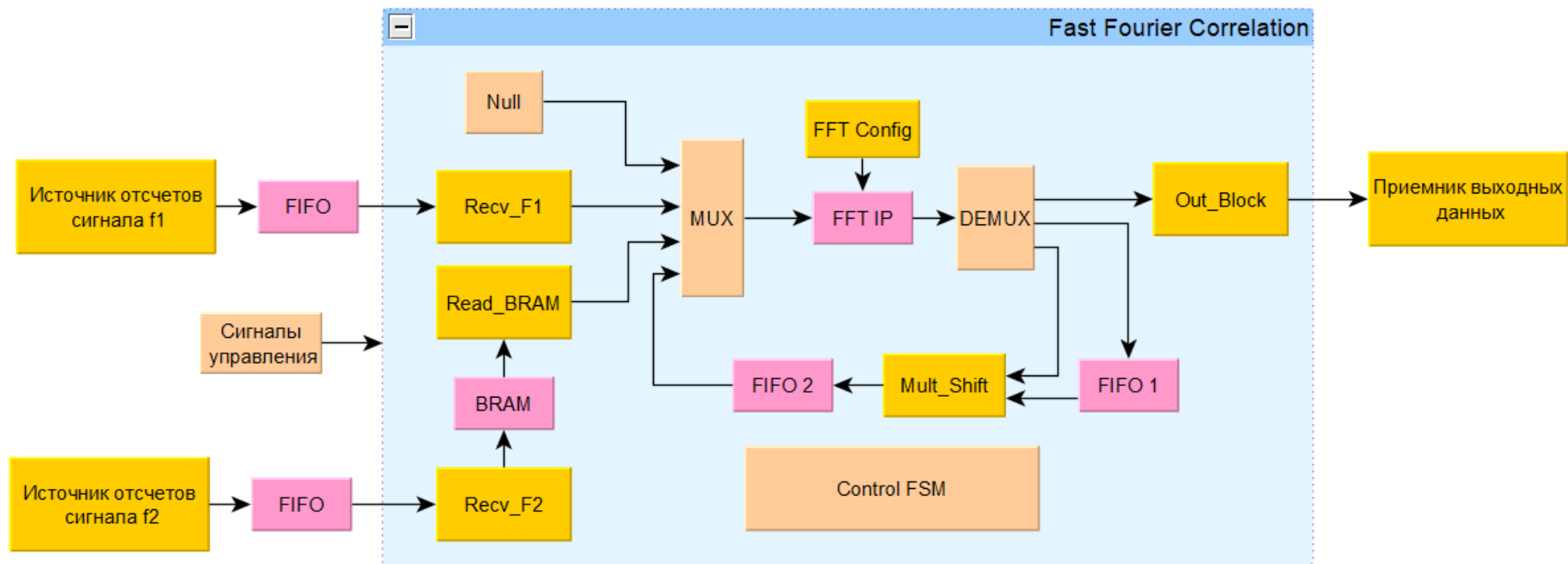


Рис. 2.1 Блок схема ядра

3. Блок конфигурации FFT IP

Блок выполняет настройку ядра Xilinx FFT на выполнение прямого или обратного преобразования Фурье. Назначение портов блока представлено в табл. 3.1.

После сброса блок находится в состоянии ожидания и сигнал `done` сброшен. При установке сигнала `start` блок переходит в рабочее состояние. В этом состоянии блок выставляет на выход `config_tdata` в зависимости от сигнала `fwd_inv` значения `0xFF` или `0x00` и устанавливает `config_tvalid`. Когда сигнал `config_tready` станет равным '1', блок установит `done` сигнал и на следующем такте перейдет в состояние ожидания. На рис. 3.1 и 3.2 представлены временные диаграммы работы и блок схема.

Таблица 3.1 Назначение портов блока

Название	Назначение
<code>aclk</code>	тактовый сигнал
<code>aresetn</code>	синхронный сигнал сброса, активный низкий уровень
<code>fwd_inv</code>	1 – прямое преобразование, 0 – обратное
<code>start</code>	сигнал начала конфигурации FFT IP
<code>done</code>	конфигурация завершена
<code>config_tdata</code>	0xFF – прямое преобразование, 0x00 – обратное
<code>config_tvalid</code>	данные установлены на шине <code>config_tdata</code>
<code>config_tready</code>	ядро FFT IP готово получить данные

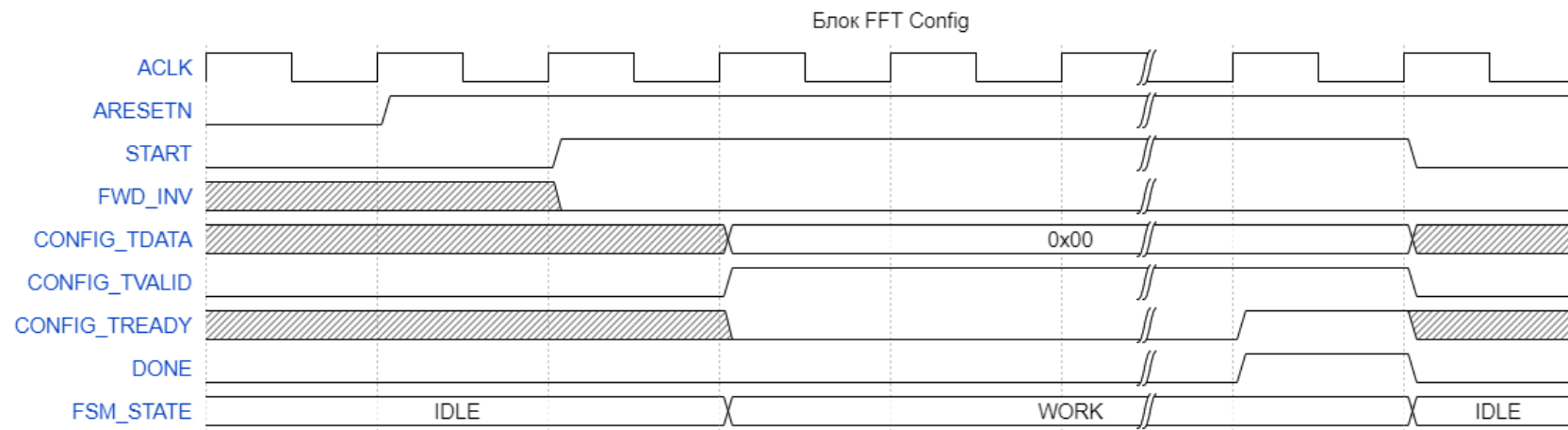


Рис. 3.1 Временные диаграммы

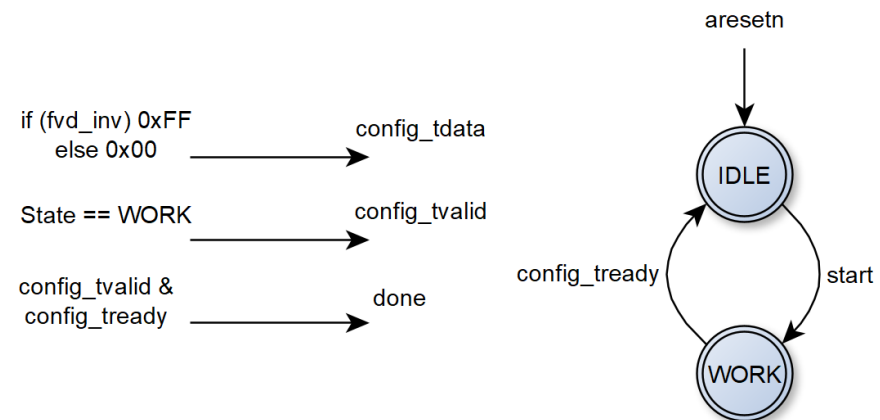


Рис. 3.2 Блок схема

4. Блок считывания сигнала f_1

Блок выполняет считывание N_1 отсчетов сигнала f_1 из FIFO на входе и выдачу их на вход мультиплексора. После выдачи заданного числа отсчетов блок выдает NFFT - N_1 нулевых отсчетов. Назначение портов блока представлено в табл. 4.1.

После сброса блок находится в состоянии ожидания. При установке сигнала *start* блок переходит в состояние считывания входных отсчетов. Когда будет считано и выдано N_1 отсчетов, блок переходит в состояние выдачи NFFT - N_1 нулевых отсчетов. Когда в сумме выдано NFFT отсчетов блок устанавливает сигнал *done* и переходит в состояние ожидания. После перехода в состояние ожидания сигнал *done* сбрасывается.

На рис. 4.1 и 4.2 представлены временные диаграммы работы и блок схема.

Таблица 4.1 Назначение портов блока

Название	Назначение
aclk	тактовый сигнал
aresetn	синхронный сигнал сброса, активный низкий уровень
N1	число считываемых отсчетов
start	сигнал начала работы
done	работа завершена
indata_tdata	отсчеты сигнала f_1
indata_tvalid	данные установлены на шине indata_tdata
indata_tready	ядро готово получить данные
outdata_tdata	выходные отсчеты
outdata_tvalid	данные установлены на шине outdata_tvdata
outdata_tready	следующее ядро готово получить данные

Таблица 4.2 Описание настраиваемых параметров

Название	Назначение
NFFT	размерность FFT

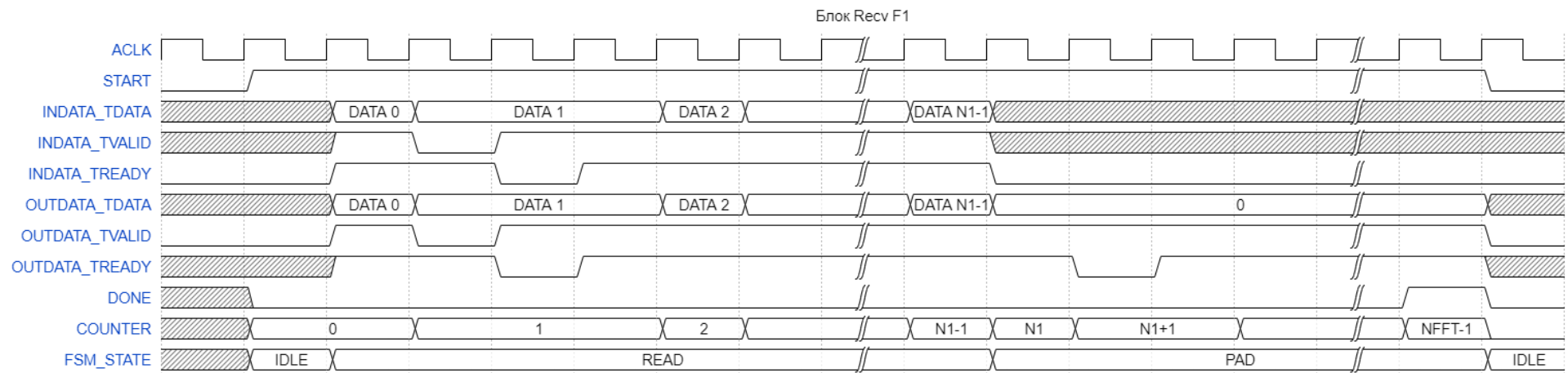


Рис. 4.1 Временные диаграммы

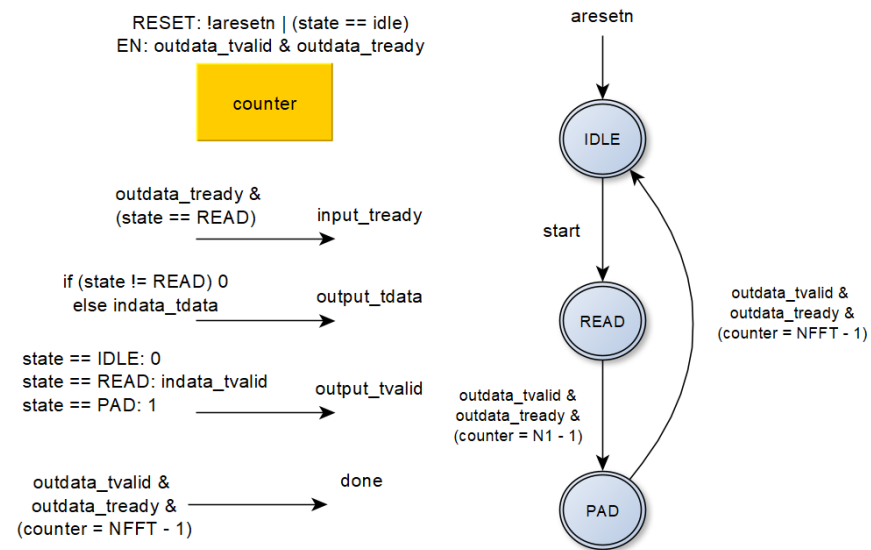


Рис. 4.2 Блок схема

5. Блок считывания сигнала f_2

Блок считывает N_2 отсчетов сигнала f_2 из FIFO на входе, выполняет комплексное сопряжение и записывает их в BRAM в обратном порядке. Назначение портов блока представлено в табл. 5.1.

После сброса блок находится в состоянии ожидания. При установке сигнала `start` блок переходит в состояние считывания входных отсчетов. Когда будет считано N_2 отсчетов, блок устанавливает сигнал `done` и переходит в состояние ожидания. После перехода в состояние ожидания сигнал `done` сбрасывается.

На рис. 5.1 и 5.2 представлены временные диаграммы работы и блок схема.

Таблица 5.1 Назначение портов блока

Название	Назначение
<code>aclk</code>	тактовый сигнал
<code>aresetn</code>	синхронный сигнал сброса, активный низкий уровень
<code>N2</code>	число считываемых отсчетов
<code>start</code>	сигнал начала работы
<code>done</code>	работа завершена
<code>indata_tdata</code>	отсчеты сигнала f_2
<code>indata_tvalid</code>	данные установлены на шине <code>indata_tdata</code>
<code>indata_tready</code>	ядро готово получить данные
<code>bram_data</code>	записываемые отсчеты
<code>bram_addr</code>	адрес записи в память
<code>bram_we</code>	разрешение записи в память

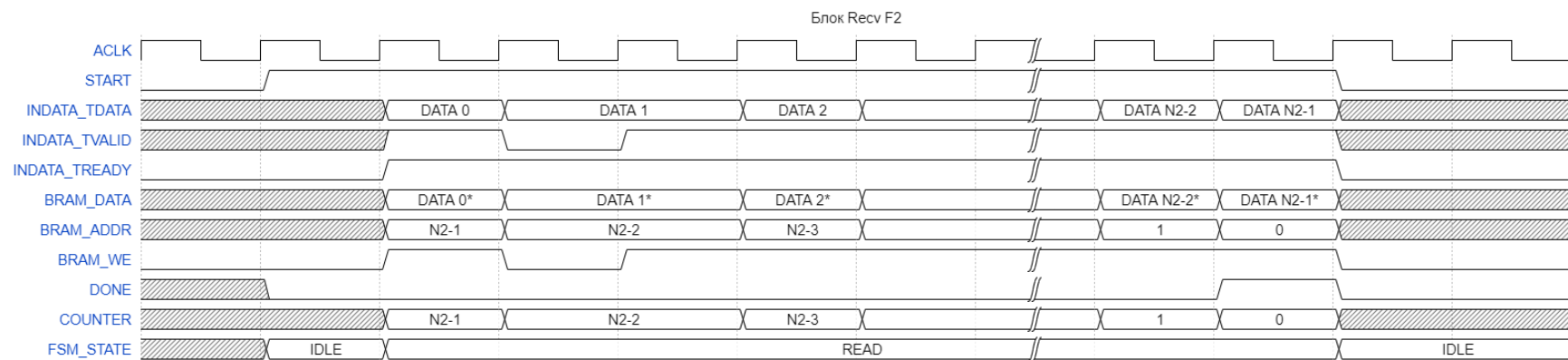


Рис. 5.1 Временные диаграммы

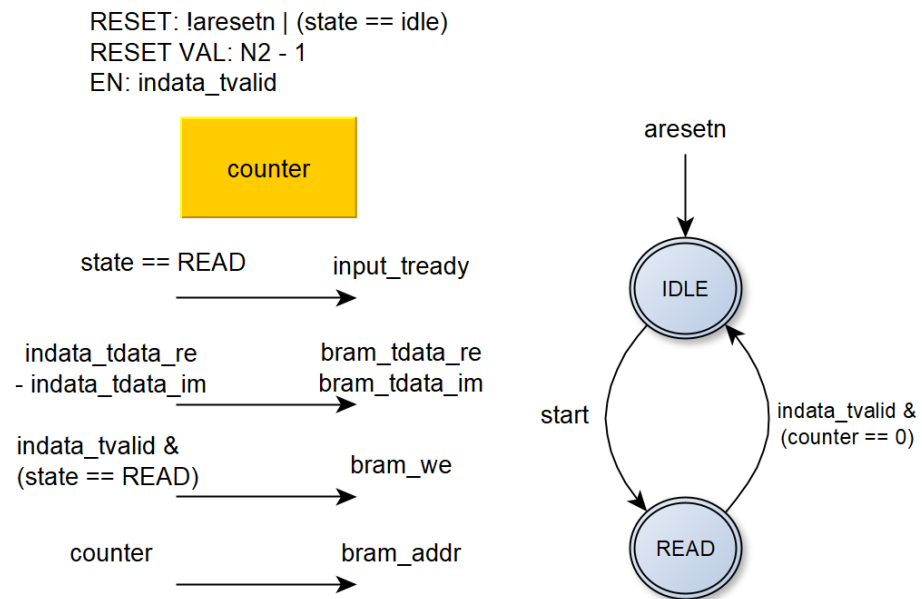


Рис. 5.2 Блок схема

6. Блок считывания BRAM

Блок выполняет считывание сопряженных отсчетов сигнала f_2 из BRAM, передает их на вход мультиплексора и дополняет нулями до заданного количества $NFFT$ отсчетов.

После сброса блок находится в состоянии ожидания. При установке сигнала `start` блок переходит в состояние считывания отсчетов из BRAM и передачи их на выход. Когда будет выдано N_2 отсчетов, блок переходит в состояние добавления нулей. Когда будет выдано $NFFT - N_2$ нулевых отсчетов блок устанавливает сигнал `done` и переходит в состояние ожидания. После перехода в состояние ожидания сигнал `done` сбрасывается.

На рис. 6.1 и 6.2 представлены временные диаграммы работы и блок схема.

Таблица 6.1 Назначение портов блока

Название	Назначение
<code>aclk</code>	тактовый сигнал
<code>aresetn</code>	синхронный сигнал сброса, активный низкий уровень
<code>N2</code>	число считываемых отсчетов
<code>start</code>	сигнал начала работы
<code>done</code>	работа завершена
<code>outdata_tdata</code>	выходные отсчеты
<code>outdata_tvalid</code>	данные установлены на шине <code>outdata_tdata</code>
<code>outdata_tready</code>	следующее ядро готово получить данные
<code>bram_data</code>	считываемые отсчеты
<code>bram_addr</code>	адрес чтения из память

Таблица 6.2 Описание настраиваемых параметров

Название	Назначение
<code>NFFT</code>	размерность FFT

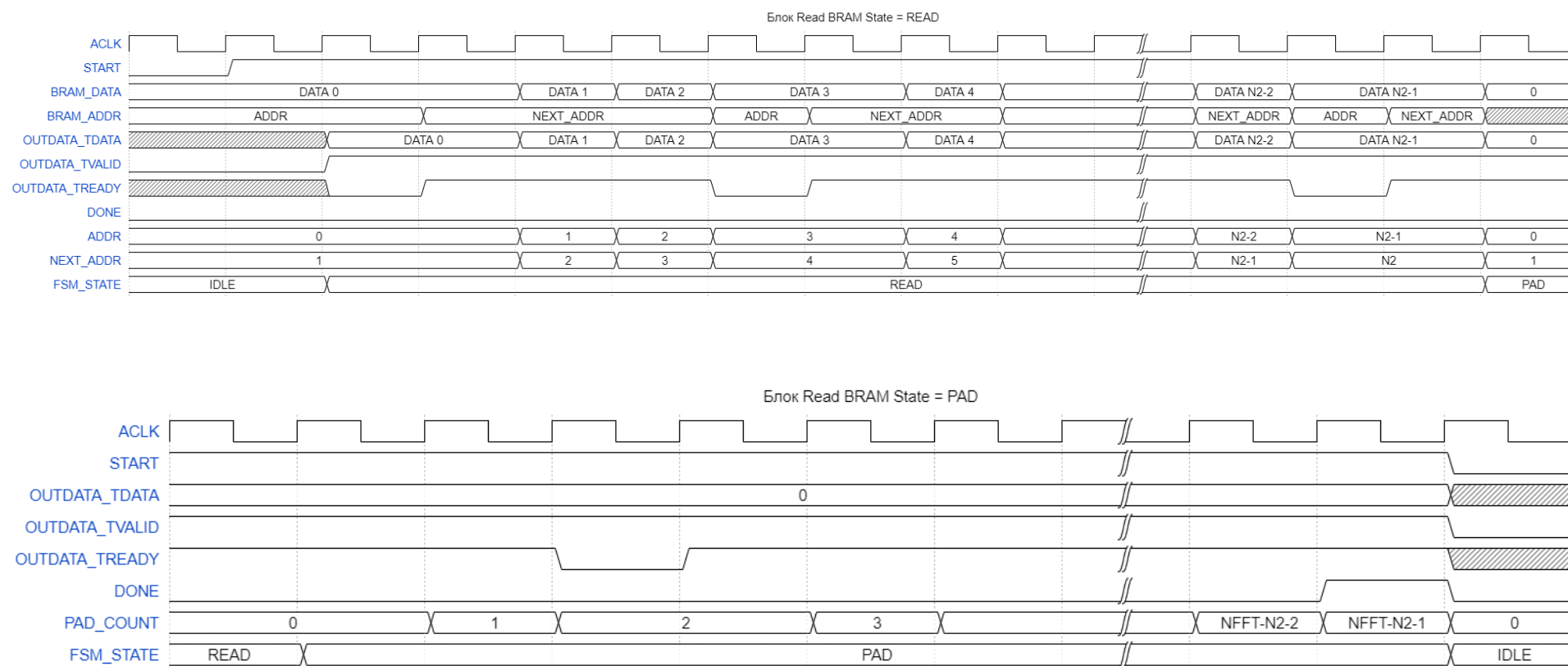


Рис. 6.1 Временные диаграммы

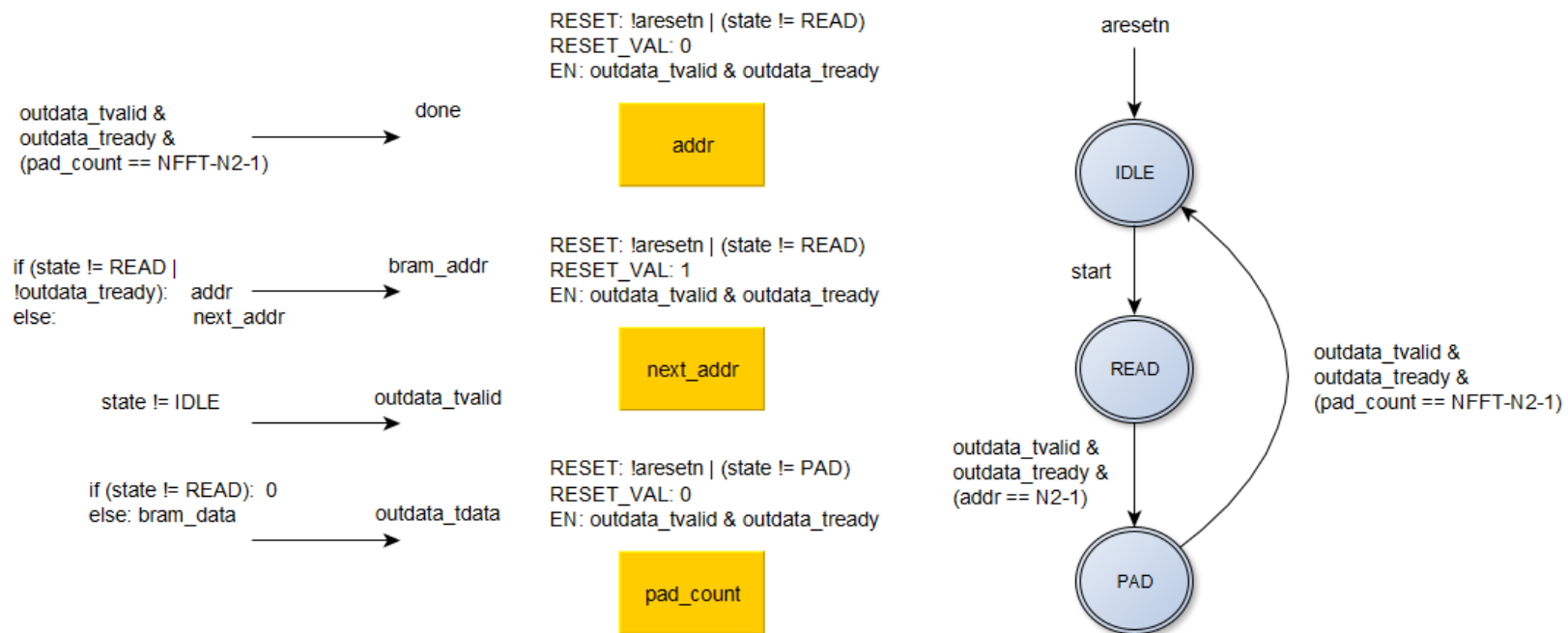


Рис. 6.2 Блок схема

7. Блок комплексного произведения и битового сдвига

Блок постоянно принимает данные из FIFO 1 и выхода демультиплексора. Когда на обоих входах появляются информационные сигналы, IP ядро Xilinx вычисляет комплексное произведение без отбрасывания бит. Далее полученный результат сдвигается на IFFT_Shift бит влево и из произведения выбираются старшие разряды. В случае переполнения выставляется флаг overflow. Выход outdata_tready отсутствует, так как FIFO 2 всегда готово принимать данные.

На рис. 7.1 представлена временные блок схема.
Таблица 7.1 Назначение портов блока

Название	Назначение
aclk	тактовый сигнал
aresetn	синхронный сигнал сброса, активный низкий уровень
IFFT_Shift	число битовых сдвигов влево
overflow	флаг переполнения
indata1_tdata	отсчеты сигнала
indata1_tvalid	данные установлены на шине indata1_tdata
indata1_tready	ядро готово получить данные
indata2_tdata	отсчеты сигнала
indata2_tvalid	данные установлены на шине indata2_tdata
indata2_tready	ядро готово получить данные
outdata_tdata	отсчеты сигнала
outdata_tvalid	данные установлены на шине outdata_tdata

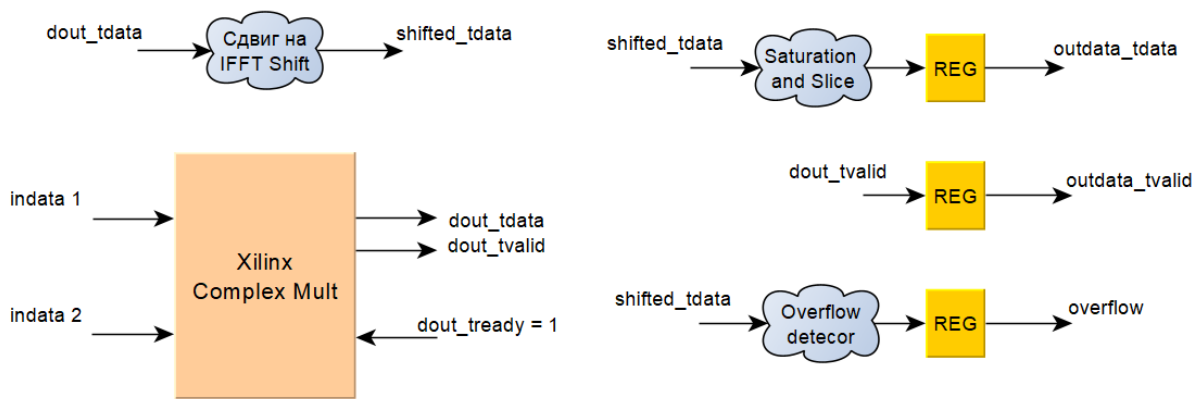


Рис. 7.1 Блок схема

8. Блок отбрасывания нулевых отсчетов

Блок выполняет считывание отсчетов обратного преобразования Фурье с выхода FFT IP, отбрасывает лишние отсчеты и выдает отсчеты корреляционной функции на выход ядра.

После сброса блок находится в состоянии ожидания. При установке сигнала `start` блок переходит в состояние выдачи отсчетов. Когда будет выдано $N_1 + N_2 - 1$ отсчетов, блок переходит в состояние отбрасывания лишних отсчетов. Когда будет отброшено $NFFT - (N_1 + N_2 - 1)$ отсчетов, блок устанавливает сигнал `done` и переходит в состояние ожидания. После перехода в состояние ожидания сигнал `done` сбрасывается.

На рис. 8.1 и 8.2 представлены временные диаграммы работы и блок схема.

Таблица 8.1 Назначение портов блока

Название	Назначение
<code>aclk</code>	тактовый сигнал
<code>aresetn</code>	синхронный сигнал сброса, активный низкий уровень
<code>N1</code>	число отсчетов сигнала f_1
<code>N2</code>	число отсчетов сигнала f_2
<code>start</code>	сигнал начала работы
<code>done</code>	работа завершена
<code>indata_tdata</code>	входные отсчеты
<code>indata_tvalid</code>	данные установлены на шине <code>indata_tdata</code>
<code>indata_tready</code>	ядро готово получить данные
<code>outdata_tdata</code>	выходные отсчеты
<code>outdata_tvalid</code>	данные установлены на шине <code>outdata_tdata</code>
<code>outdata_tready</code>	следующее ядро готово получить данные

Таблица 8.2 Описание настраиваемых параметров

Название	Назначение
<code>NFFT</code>	размерность FFT

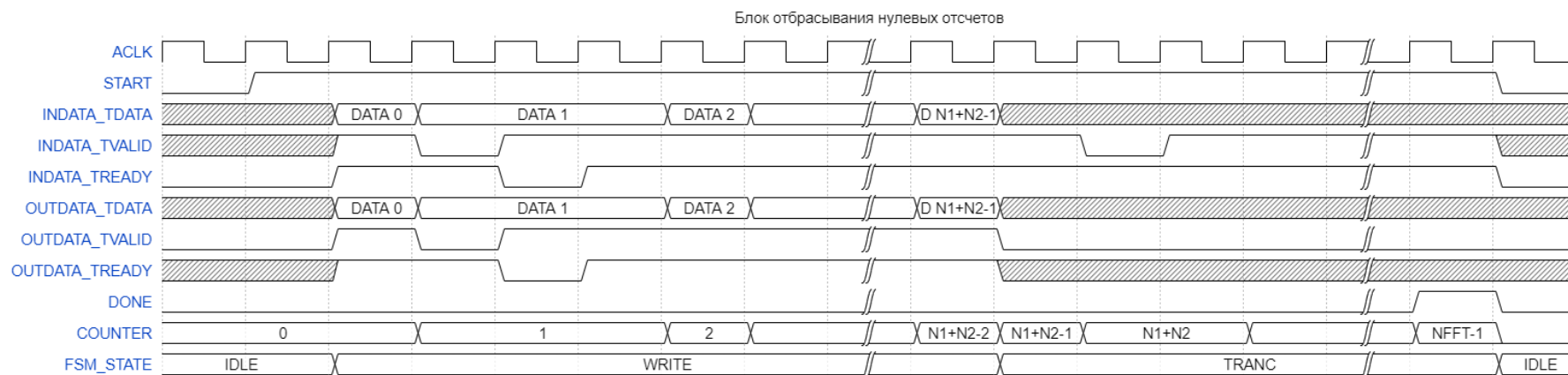


Рис. 8.1 Временные диаграммы

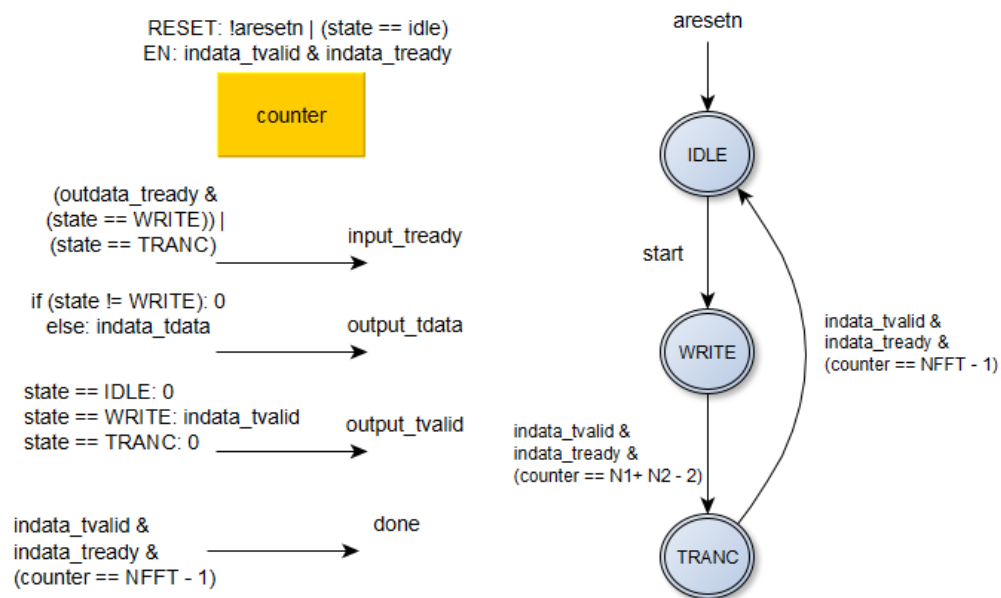


Рис. 8.2 Блок схема

9. Конечный автомат управления блоками ядра

Блок задает очередность выполнения других блоков ядра, формируя сигналы `start` и считывая сигналы `done`. После сброса блок находится в состоянии ожидания. При поступлении сигнала `start` блок запускает блок `FFT_Config` для настройки ядра `FFT_IP` в режим прямого преобразования.

Далее запускаются блоки `Recv_F1` и `Recv_F2`. Дождавшись завершения работы блоков `Recv_F2` и `FFT_IP`, блок управления стартует блок `Read_BRAM`. После завершения работы блока `FFT_IP` запускается блок `FFT_Config` для настройки `FFT_IP` в режим обратного преобразования.

После этого блок управления стартует блок `Out_Block`. После завершения его работы автомат переходит в состояние ожидания сигнала `start`. На рис. 9.1 представлена блок схема.

Таблица 9.1 Назначение портов блока

Название	Назначение
<code>aclk</code>	тактовый сигнал
<code>aresetn</code>	синхронный сигнал сброса, активный низкий уровень
<code>start</code>	запуск расчета корреляционной функции
<code>fwd_inv</code>	прямое или обратное преобразование
<code>FFT_IP_tlast</code>	сигнал завершения работы ядра <code>FFT_IP</code>
<code>FFT_Config_Start</code>	старт блока <code>FFT_Config</code>
<code>FFT_Config_Done</code>	завершение работы блока <code>FFT_Config</code>
<code>Recv_F1_Start</code>	старт блока <code>Recv_F1</code>
<code>Recv_F2_Start</code>	старт блока <code>Recv_F2</code>
<code>Recv_F2_Done</code>	завершение работы блока <code>Recv_F2</code>
<code>Read_BRAM_Start</code>	старт блока <code>Read_BRAM</code>
<code>Read_BRAM_Done</code>	завершение работы блока <code>Read_BRAM</code>
<code>Out_Block_Start</code>	старт блока <code>Out_Block</code>
<code>Out_Block_Done</code>	завершение работы блока <code>Out_Block</code>
<code>Mux_Sel</code>	выбор канала мультиплексора
<code>Demux_Sel</code>	выбор канала демультиплексора

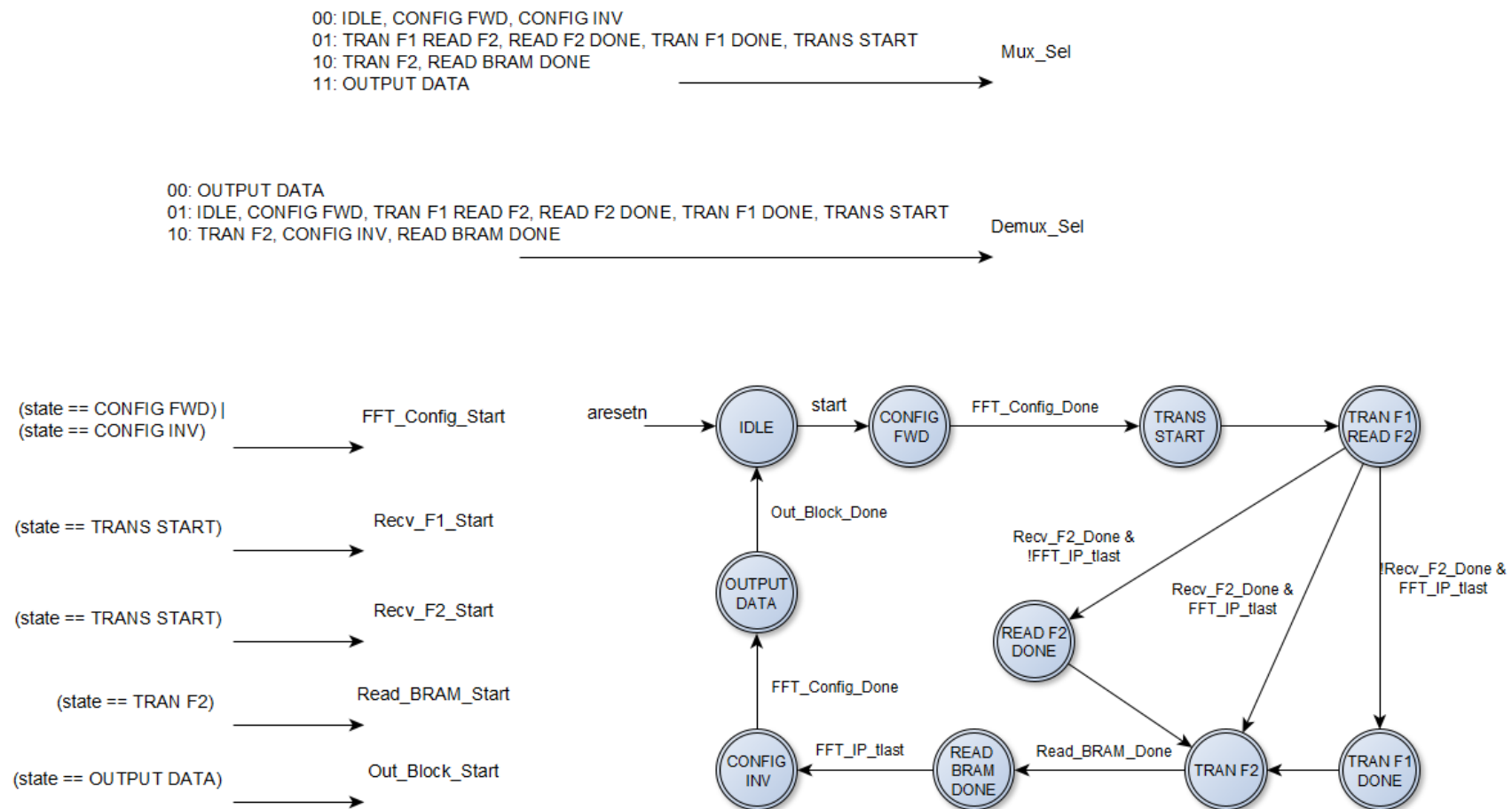


Рис. 9.1 Блок схем