

Методические указания к выполнению контрольной работы №3

1. Теоретическая часть

Данная контрольная работа направлена на развитие навыков отображения RTL-моделей цифровых блоков, написанных на языке Verilog HDL, на ресурсы микросхем программируемой логики (ПЛИС).

Поясним этапы, которые проходит проект цифровой схемы, начиная от создания спецификации схемы на языке Verilog HDL и заканчивая прошивкой ПЛИС. Выделяют следующие этапы:

- Функциональная спецификация на языке RTL-уровня (в нашем случае на Verilog HDL).
- Функциональная (поведенческая) симуляция (Behavioral simulation) – позволяет провести тестирование RTL-модели без учета временных задержек реальных элементов. Проверяется алгоритм работы схемы.
- Логический синтез (RTL синтез) – на данном этапе спецификация RTL преобразуется в модель уровня цифровых вентилей.
- Ввод ограничений в проект – наложение ограничений на модель, которые учитывают особенности конечной аппаратной платформы, а также вопросы синхронизации. Определяется соответствие виртуальных портов модели реальным портам ПЛИС, указывается период реального входного тактового сигнала.
- Отображение (mapping) – синтезированная RTL-модель переводится в элементную базу конечной аппаратной платформы – конфигурируемые макро-ячейки ПЛИС (для ПЛИС Xilinx это CLB блоки).
- Трассировка и размещение (place and route) – размещение на элементной базе ПЛИС с учетом введенных ограничений.
- Статический временной анализ (static timing analysis) – проверка результата предыдущего шага на удовлетворение ограничениям. На данном шаге находится путь в схеме с максимальной задержкой и рассчитывается максимальная частота работы схемы. Если максимальная частота меньше, чем частота входного тактового сигнала, то выполняется повтор предыдущего или предыдущих двух шагов. Если в течении нескольких итераций не удастся найти размещение, удовлетворяющее ограничением, то выводится соответствующая ошибка в используемом САПР (в нашем случае в Vivado Design Suite).
- Временная симуляция (Post-Route Simulation) – симуляция с учетом задержек на элементах аппаратной платформы и связях между ними. Проводится редко. Проводится обычно, когда поведенческая симуляция не позволяет выявить ошибку.
- Генерация конфигурационного файла.
- Конфигурация ПЛИС и отладка на «железе».

Рассмотрим подробнее процесс проверки временных ограничений.

1.1. Временные ограничения

Существует проблема, связанная с комбинационными схемами (КС). Данная проблема заключается в том, что в неидеальной комбинационной схеме существуют задержки на ее элементах. Из-за этого в момент переключения схемы значение на ее выходе может быть произвольным, о достоверности которого трудно судить. Возьмем, к примеру, двухвходовой мультиплексор.

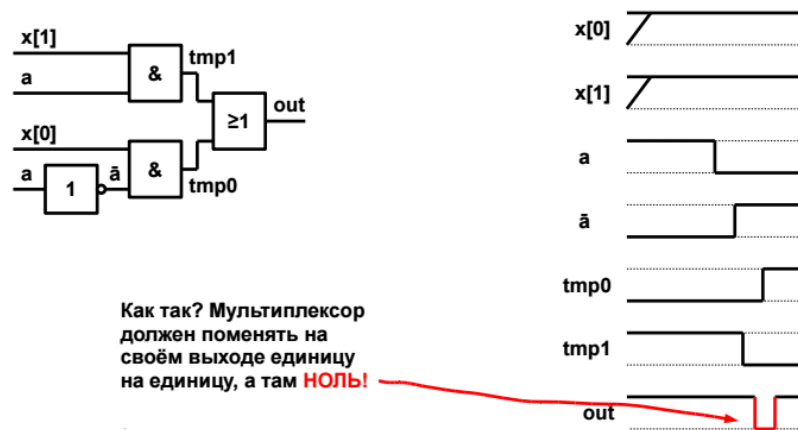


Рисунок 1. Неидеальный двухвходовой мультиплексор.

Для решения этой проблемы необходимо ввести источник событий, моменты наступления которых гарантируют достоверность всех сигналов на выходе комбинационной схемы. В качестве такого источника событий служит тактовый сигнал (clock, clk), а сами события называются фронтами (edges). Эти фронты используются как сигналы разрешения работы элементов памяти, управляемых по фронту – синхронных триггеров (flip-flop) или регистров.

Тактовый сигнал характеризуется следующими параметрами:

- Период (частота).
- Коэффициент заполнения (величина, обратная параметру скважности) – duty cycle.
- Скорость нарастания фронта (slew rate).
- Сдвиг фазы (clock skew).
- Дрожание фазы (jitter).

Последовательные схемы, содержащие КС и элементы памяти с управлением фронтом общего тактового сигнала, называются синхронными. Все прочие схемы называются асинхронными. В зависимости от количества используемых в синхронизации схемы фронтов, различают схемы с синхронизацией:

- По переднему или заднему фронту – Single data rate (SDR).
- По обоим фронтам – Double data rate (DDR).
- По обоим фронтам двух тактовых сигналов, сдвинутых по фазе на 90 градусов друг относительно друга – Quad data rate (QDR).

Можно сказать, что в синхронных схемах все элементы памяти получают один и тот же фронт сигнала одновременно, однако это не так. Каждый элемент памяти получает этот

фронт с некоторым сдвигом фазы относительно его генератора (поскольку время распространения конечно). Таким образом, можно говорить о том, что между двумя элементами памяти существует сдвиг фазы тактового сигнала (T_{skew}).

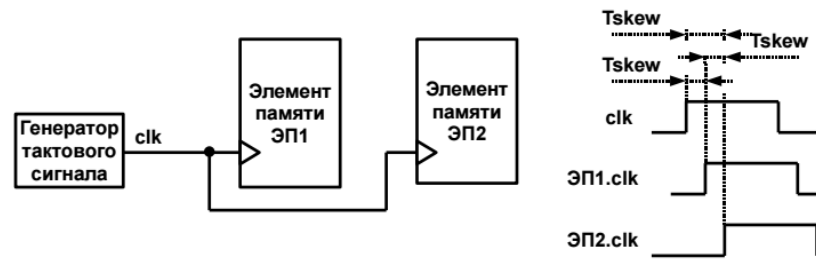


Рисунок 2. Сдвиг фазы между двумя элементами памяти.

На рисунке 2 показана временная диаграмма, на которой отмечены три различных сдвига фазы: для каждого элемента памяти относительно генератора, а также между элементами памяти.

Помимо T_{skew} в синхронных схемах существуют и другие возможные ограничения. Рассмотрим синхронный D-триггер, который является основой синхронных схем (рисунок 3).

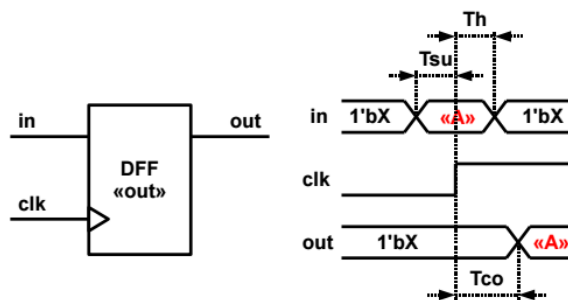


Рисунок 3. Синхронный D-триггер и временная диаграмма его работы.

На рисунке 3 продемонстрирована работа синхронного D-триггера в момент его первой записи. Здесь T_{su} (T_h) – минимальное время, которое входные данные должны удерживаться на входе перед фронтом (после фронта), чтобы быть защелкнуты триггером по переднему фронту (предустановка – setup и удержание – hold соответственно). T_{co} – максимальное время от факта подачи фронта синхросигнала до появления «защелкнутых» данных на выходе триггера.

1.1.1. Синхронная схема с одним тактовым доменом

На рисунке 4 показана схема синхронной очереди с дисциплиной FIFO. В данном случае $T_{skew} = 0$. На рисунке 4 отмечены интервалы предустановки и удержания. Используется один тактовый сигнал, а следовательно все элементы схемы находятся в одном тактовом домене.

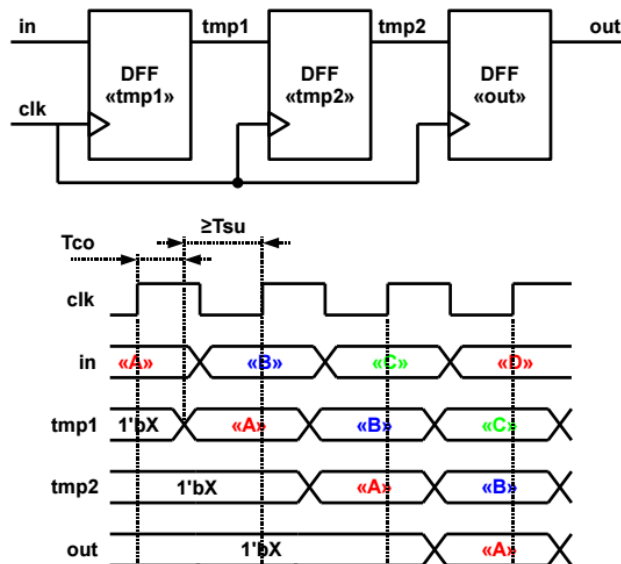


Рисунок 4. Простая синхронная схема.

1.1.1. Синхронная схема и несколько тактовых доменов

На рисунке 5 показана схема, использующая два разных генератора тактовых сигналов, T_{skew} между которыми неизвестно. В этом случае, если $T_{skew} < T_{co} + T_{su}$, то триггер «out» нарушит свой интервал предустановки и перейдет в метастабильное состояние (состояние, в котором он будет иметь случайное значение на выходе). Такая ситуация продемонстрирована на временной диаграмме на рисунке 5.

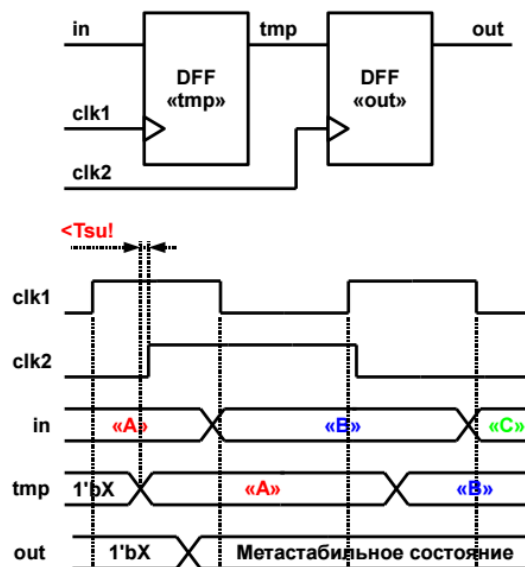


Рисунок 5. Простая асинхронная схема.

Асинхронности в схемах стоит избегать. Однако существуют ситуации, в которых используются так называемые домены синхронизации (clock domain). Домен синхронизации – часть схемы с общим тактовым сигналом. В каждой микросхеме может быть несколько таких доменов. Обычно этот термин относится к рассмотрению ПЛИС в качестве таких микросхем. Такие схемы называются глобально асинхронными – локально синхронными (GALS).

1.1.2. Типичная синхронная схема и критерии ее работоспособности

Рассмотрим схему, описанную с помощью модели на RTL уровне (рисунок 6).

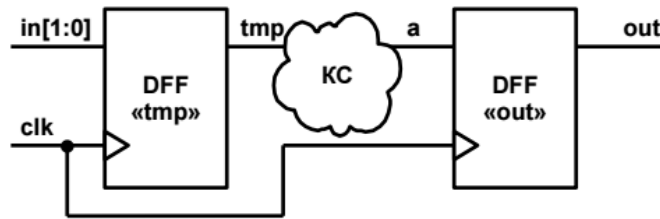


Рисунок 6. Типичная синхронная схема.

На рисунке 7 приведена временная диаграмма работы такой схемы.

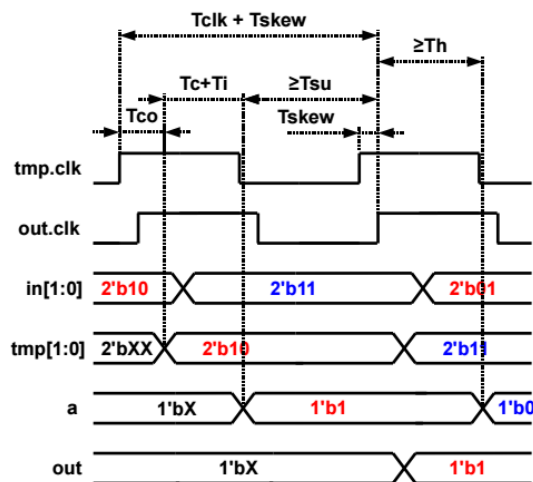


Рисунок 7. Временная диаграмма работы типичной синхронной схемы.

На рисунке 7 показаны интервалы предустановки и удержания регистров, сдвиг фаз между фронтами их работы, а также следующие параметры:

- T_c – задержка в комбинационной схеме (время, через которое выход КС станет достоверным).
- T_i – задержка на проводах.
- T_{clk} – период тактового сигнала.

Параметры T_{co} , T_{su} , T_h – характеристики микросхемы, которые зависят от техпроцесса, температурного диапазона, напряжения питания и т.д.

Параметры T_c и T_i зависят от результатов отображения, размещения и трассировки, как следствие их можно варьировать, изменяя стратегию соответствующих процессов.

T_{skew} зависит от результатов трассировки тактового сигнала.

T_{clk} – требуемый период тактового сигнала. Величина периода определяется двумя способами:

- Устанавливается требуемый период, после чего величины T_c и T_i подгоняются под него (путем итеративного повторения шагов отображения, размещения и

трассировки, либо изменениями в исходной спецификации RTL модели на языке HDL, например Verilog HDL) – оптимизация схемы.

- Определение периода по текущим значениям T_c и T_i .

Учитывая данные параметры, можно определить *критерий работоспособности* синхронной схемы, при нарушении которого триггер может защелкнуть недостоверное значение или вообще перейти в метастабильное состояние.

$$T_{clk} + T_{skew} \geq T_{co} + T_c + T_i + T_{su}$$

$$T_{co} + T_c + T_i + T_{su} \geq T_{skew} + T_h$$

Невыполнение первого критерия называется нарушением времени предустановки триггера (setup violation) – данные приходят на триггер слишком близко к фронту тактового сигнала.

Невыполнение второго критерия называется нарушением времени удержания (hold violation) – фронт тактового сигнала, по которому данные были выставлены триггером-источником, защелкивает их в триггер-приемник, хотя защелкивать их должен следующий фронт.

T_{skew} больше нуля, когда тактовый сигнал раньше приходит на регистр-источник данных, и меньше нуля, когда тактовый сигнал раньше приходит на регистр-приемник данных.

Опираясь на критерий работоспособности, можно рассчитать быстродействие синхронной схемы – максимальную частоту (минимальный период – T_{clk}), на которой может работать схема.

Задача определения соответствия схемы критерию работоспособности называется задачей статического временного анализа (Static timing analysis). В САПР Vivado Design Suite эта задача решается автоматически, наряду с задачами синтеза RTL, отображения и PAR (place and route). Как данность берутся параметры T_{su} , T_h , T_{co} , T_{lut} (задержка в LUT) и т.д., определяемые библиотекой для выбранной микросхемы ПЛИС. После выполнения PAR будут подсчитаны величины T_i и T_{skew} , после чего будет определено, соответствует ли схема критерию работоспособности и целевому T_{clk} , либо будет предложено минимальное значение T_{clk} , если этот параметр не был заранее установлен.

2. Пример решения задания контрольной работы

Дана спецификация модели на языке Verilog:

```
module FSM (  
    input      clk, reset,  
    input  [1:0] a,  
    output [2:0] y  
);  
  
    reg [1:0] state;  
  
    assign y = (state == 2'b00) ? 3'b001 :  
               (state == 2'b01) ? 3'b010 :  
               (state == 2'b10) ? 3'b100 : 3'b000;  
  
    always @(posedge clk, posedge rst) begin  
        if(rst)  
            state <= 0;  
        else case(state)  
            2'b00: if ( a[1]) state <= 2'b01;  
            2'b01: if ( a[0]) state <= 2'b10;  
            2'b10: if (!a[0]) state <= 2'b01;  
                else if ( a[1]) state <= 2'b00;  
        endcase  
    end
```

Необходимо выполнить следующие шаги:

- провести синтез RTL модели;
- определить необходимое количество конфигурируемых ячеек ПЛИС, необходимое для размещения RTL модели и выполнить непосредственно размещение (структура ячейки показана на рисунке 8 и состоит из двух LUT и двух регистров/flip-flop);
- выполнить статический временной анализ:
 - определить критерии работоспособности получившейся схемы;
 - определить быстродействие получившейся схемы.

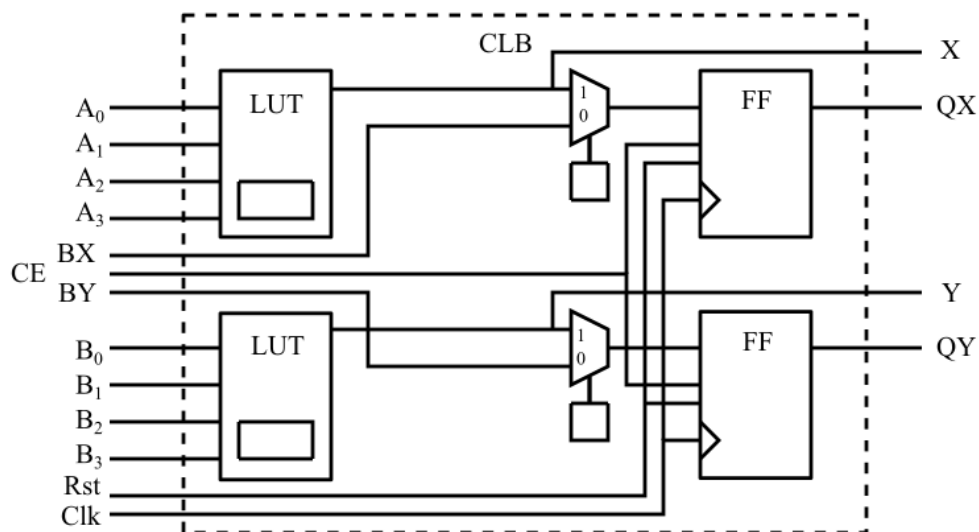


Рисунок 8. Внутренняя структура CLB.

Временные характеристики элементов, входящих в состав CLB:

- $T_{lut} = 0.36$ нс (задержка в блоках LUT)
- $T_{su} = 1.88$ нс
- $T_h = 0.13$ нс
- $T_{co} = 0.68$ нс
- $T_{i0} = 0.09$ нс (задержка на мультиплексоре)
- $T_{i1} = 0.01$ нс (задержка в обход мультиплексора)

Задержками, возникающими на внешних связях, а также разностью фаз тактовых сигналов пренебречь (принять как равными нулю).

3. Решение.

Как результат необходимо получить две схемы (как результат RTL синтеза и размещения), а также рассчитанные значения критериев работоспособности и максимальной частоты работы схемы (как результат статического временного анализа).

3.1. RTL синтез.

Задача RTL синтеза заключается в получении RTL модели из спецификации на языке HDL (Verilog). В спецификации, приведенной выше, можно явно выделить описание запоминающего элемента, а также двух комбинационных схем. Для синтеза комбинационных схем нужно составить соответствующие таблицы истинности. В данном случае выходами этих схем будут выходные сигналы схемы – $y[2:0]$ (y_0, y_1, y_2), а также сигналы, формирующие следующее состояние. Последние явно в спецификации не обозначены, поэтому в дальнейшем примем эти сигналы за $next_state[1:0]$ (ns_0 и ns_1).

Составим таблицы истинности для каждой группы сигналов (таблица 1 и таблица 2).

state[1] (s_1) / state[0] (s_0) / a[1] (a_1) / a[0] (a_0)	ns_1	ns_0
0/0/0/0	0	0
0/0/0/1	0	0
0/0/1/0	0	1
0/0/1/1	0	1
0/1/0/0	0	1
0/1/0/1	1	0
0/1/1/0	0	1
0/1/1/1	1	0
1/0/0/0	0	1
1/0/0/1	1	0
1/0/1/0	0	1
1/0/1/1	0	0
1/1/0/0	X	X

1/1/0/1	X	X
1/1/1/0	X	X
1/1/1/1	X	X

Таблица 1. Логика формирования следующего состояния.

state[1] (s_1)	state[0] (s_0)	y[2:0]
0	0	001
0	1	010
1	0	100
1	1	000

Таблица 2. Логика формирования выхода автомата.

Основываясь на таблице 1, можно записать:

$$ns_0 = \bar{s}_1 \bar{s}_0 a_0 \vee \bar{s}_1 s_0 \bar{a}_0 \vee \bar{s}_0 s_1 \bar{a}_0$$

$$ns_1 = \bar{s}_1 s_0 a_0 \vee \bar{s}_0 s_1 \bar{a}_1 a_0$$

На основе этих двух выражений можно построить комбинационную схему. Также можно минимизировать количество используемых элементов посредством выделения общих частей в выражениях. Комбинационная схема логики следующего состояния приведена на рисунке 9.

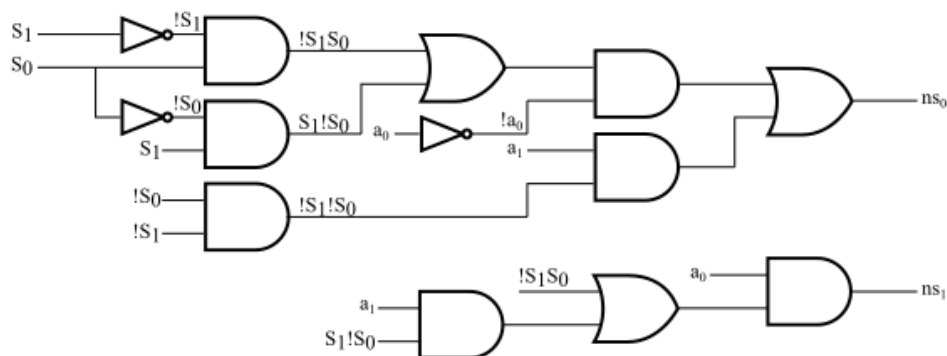


Рисунок 9. Комбинационная схема логики следующего состояния.

Стоит обратить внимание, что схема на рисунке 9 имеет четыре входных сигнала и два выходных в соответствии с таблицей 1. Аналогичным образом можно получить комбинационную схему для выходных сигналов автомата (рисунок 10).

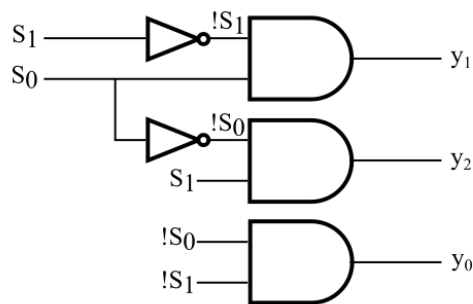


Рисунок 10. Комбинационная схема логики формирования выходных сигналов.

Стоит отметить, что выходные сигналы можно было вывести из комбинационной схемы, изображенной на рисунке 9, соответствующей логики следующего состояния. На различных схемах они показаны, потому что в дальнейшем отображении на CLB ПЛИС эти комбинационные схемы будут располагаться физически в разных местах. При реализации данных комбинационных схем на жесткой логике целесообразней было бы вывести сигналы $y[2:0]$ из комбинационной схемы логики следующего состояния.

Оставшийся регистр можно изобразить с помощью стандартного базового операционного элемента (БОЭ) (рисунок 11).

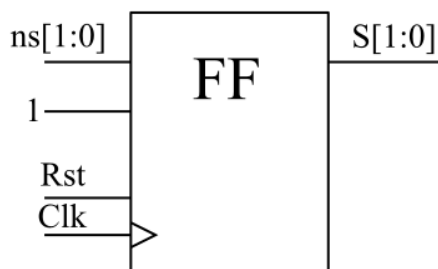


Рисунок 11. Регистр, хранящий состояние автомата.

На рисунке 11 вход разрешения записи регистра всегда имеет значение логической единицы. Далее необходимо свести получившиеся схемы в единую систему (рисунок 12), для чего можно изобразить ее функциональную схему.

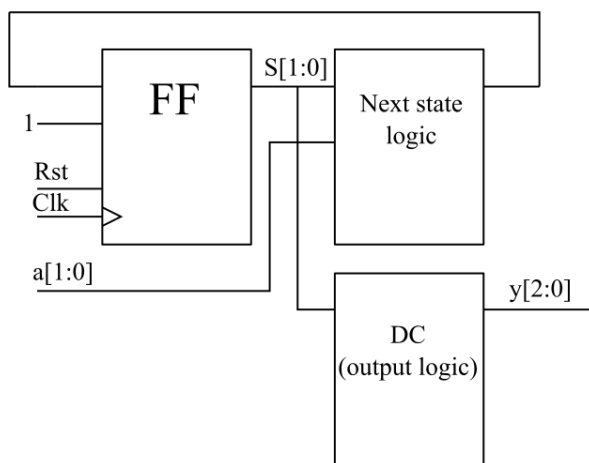


Рисунок 12. Функциональная схема конечного автомата на RTL уровне.

Результат RTL синтеза.

Таким образом, результатом RTL синтеза модели, описанной на языке Verilog, можно назвать схемы, представленные на рисунках 9 – 12.

3.2. Задача размещения.

Чтобы реализовать получившуюся схему на ПЛИС, ее нужно преобразовать с использованием элементной базы ПЛИС (выполнить этап отображения). Этой элементной базой являются конфигурируемые ячейки (CLB), которые включают в свой состав LUT и синхронные регистры (flip-flop). Помимо компонент, необходимых для реализации некоторой логики в состав CLB также входят компоненты, позволяющие конфигурировать связи между внутренними элементами, а также между этими элементами и выходами. Структура подобной ячейки приведена на рисунке 8.

LUT – таблица поиска (look-up table) – представляет собой комбинационную схему с определенным количеством входов, которую можно конфигурировать, чтобы получать произвольную функцию булевой алгебры, зависящую от определенного количества аргументов, значением которой является одно-битное слово. LUT строится на мультиплексорах. На рисунке 6 показана схема LUT «4-в-1». Она представляет собой каскад мультиплексоров, адресные входы которых являются входами LUT (в данном случае их 4), причем ближе к началу каскада на адресные входы подаются наиболее значимые биты входного слова. Входы первых в каскаде мультиплексоров заведены на некоторую память (в ПЛИС это энергозависимая память), в которой хранятся значения соответствующей функции в таблице истинности (верхний бит соответствует нижнему биту в таблице истинности). При прошивке ПЛИС в эту память записывается необходимое значение, после чего LUT начинает работать как конкретная функция от 4 входов, выдавая 1 бит результата на выходе X.

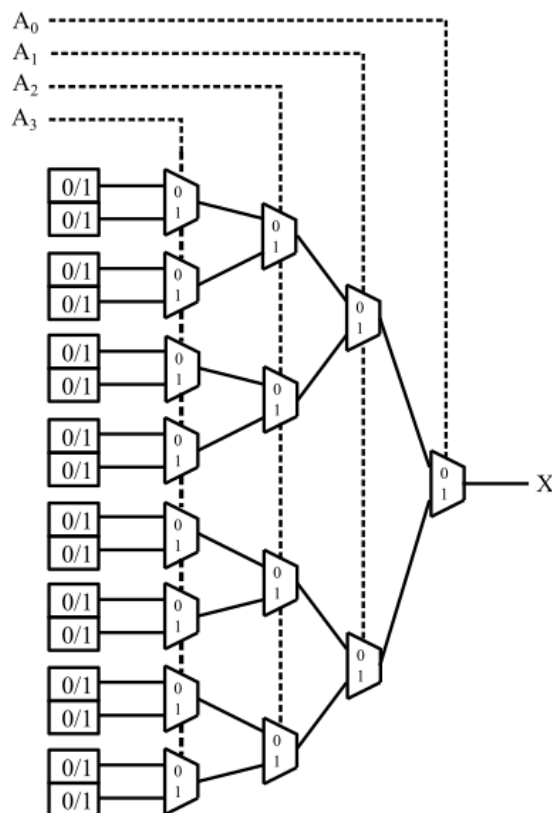


Рисунок 13. Устройство LUT «4-в-1».

В нашем случае у нас есть два сигнала, которые соответствуют следующему состоянию ($ns[1:0]$), а также три сигнала, соответствующие выходам автомата ($y[2:0]$). Для функций сигналов следующего состояния таблица истинности была представлена в таблице 1. Для ее реализации в контексте ПЛИС необходимо взять 2 LUT, что можно сделать с использованием одной конфигурируемой ячейки (рисунок 14).

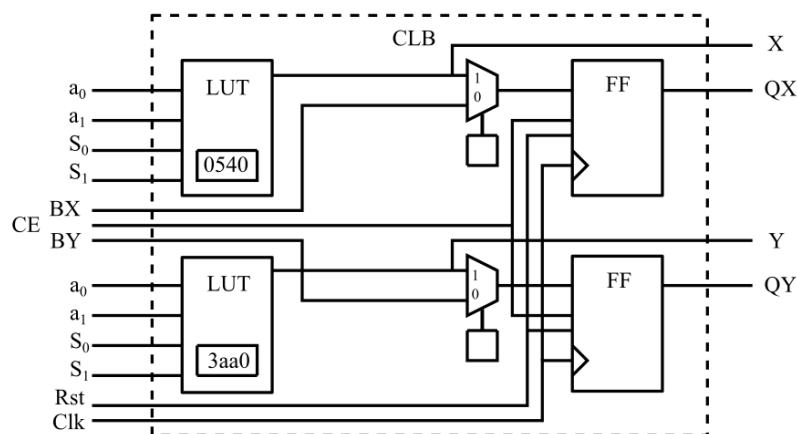


Рисунок 14. Размещение логики следующего состояния.

Как можно заметить, на рисунке 14 изображена конфигурируемая ячейка, LUT которой содержат значения, соответствующие значениям таблицы истинности для сигнала ns_0 (нижний LUT) и сигнала ns_1 (верхний LUT). Эти значения записаны в шестнадцатеричном виде, а значению X в таблице истинности было сопоставлено значение 0 (поскольку эти значения не имеют смысла и не могут возникнуть в ходе работы системы).

Выше было сказано, что CLB содержат коммутационные элементы. На рисунках 8 и 14 этими элементами являются два мультиплексора, на входы каждого из которых подается пара сигналов – выходной сигнал LUT и один из сигналов BX и BY соответственно. Данные мультиплексоры, как и LUT конфигурируются на этапе прошивки, чтобы была возможность подавать значение в регистр в обход LUT. В нашем случае следующее состояние, формирующееся в одном CLB, можно сохранить при помощи регистров, располагающихся в этой же CLB. Изобразим это на рисунке 15, опустив при этом все неиспользуемые сигналы (BX , BY , X , Y), заведя сигнал CE на питание (регистр должен сохранять значение каждый такт), а также переименовав условные обозначения сигналов в обозначения из спецификации ($XQ - S_0$, $YQ - S_1$).

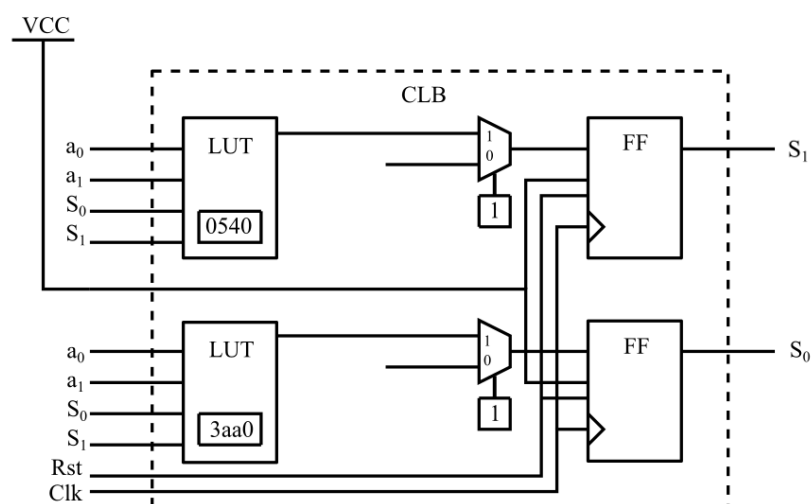


Рисунок 15. Размещение логики формирования следующего состояния и памяти состояния.

На рисунке 15 показана CLB ячейка, в которой располагается логика формирования следующего состояния, а также память состояния (регистр). Осталось расположить логику формирования выхода. Для этого необходимо обратиться к таблице 2, в которой приведены таблицы истинности для сигналов $y[2:0]$. По аналогии с сигналами $ns[1:0]$ можно задействовать LUT. Однако, каждый сигнал y_i формируется на базе двух входных сигналов. Кроме того, таких сигналов всего три, из-за чего необходимо взять три LUT, или две CLB.

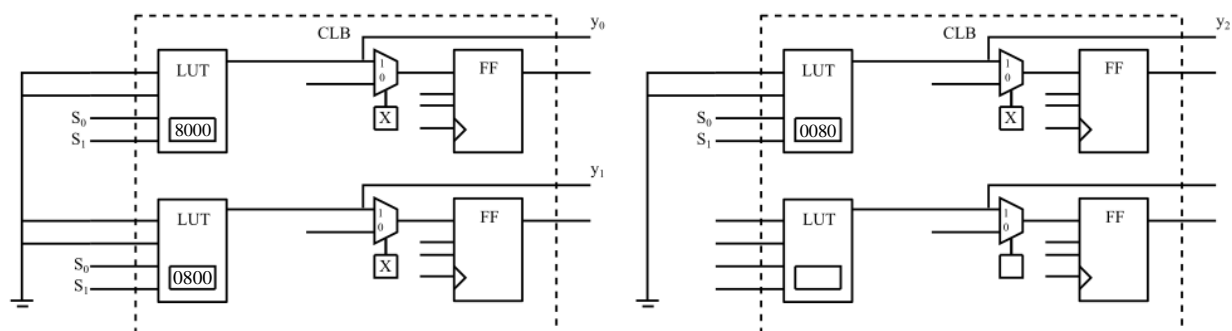


Рисунок 16. Размещение комбинационной логики выходов.

Поскольку в LUT с четырьмя входами будут задействованы всего два входа, необходимо подать логический ноль на оставшиеся, чтобы «исключить» неиспользуемые мультиплексоры из каскада (на рисунке 13 это нижние семь мультиплексоров). Соответственно на используемых входах мультиплексоров могут быть любые значения. Получившиеся конфигурации CLB изображены на рисунке 16. Стоит обратить внимание, что на адресных входах коммутационных мультиплексоров находится значение «X», это значит, что оно нам не важно, поскольку блоки FF в показанных CLB не используются.

Результат процесса размещения.

Таким образом, результатом размещения являются схемы, изображенные на рисунках 15 – 16, с учетом записанных значений в ячейки памяти LUT и адресных входов коммутирующих мультиплексоров.

3.3. Задача статического временного анализа.

В пункте 1 были описаны вводимые ограничения на работу синхронной микросхемы. Эти ограничения заключаются в необходимости выполнения критерия работоспособности, приведенного в 1.1.3, при заданной частоте (периоде T_{clk}), либо с получением значения максимальной частоты работы.

Для удобства еще раз запишем критерий работоспособности:

$$T_{clk} + T_{skew} \geq T_{co} + T_c + T_i + T_{su}$$

$$T_{co} + T_c + T_i + T_{su} \geq T_{skew} + T_h$$

В данном случае есть два синхронных регистра на основе D-триггера. Параметры для расчета критерия для каждого из них одинаковы в данном задании, что следует из рисунка 15. Здесь $T_c = T_{lut}$, а $T_i = T_{i0}$, что также следует из рисунка 15. $T_{skew} = 0$ по условию. Остальные параметры заданы. В итоге можно получить:

$$T_{clk} \geq 0.68 + 0.36 + 0.09 + 1.88 \geq 0.13$$

$$T_{clk} \geq 3.01 \geq 0.13$$

Результат статического временного анализа.

Таким образом, каждый из используемых синхронных регистров способен работать от тактового сигнала с минимальным периодом в 3.01 нс, что соответствует частоте (максимально возможной) 333 МГц.