

Продолжение. Начало в № 3 `2008

Иосиф КАРШЕНБОЙМ  
iosifk@narod.ru

### Краткий обзор

Сейчас только самые элементарные логические схемы используют одну синхрочастоту. Большинство современных устройств, работающих с данными, включая микроконтроллеры, модемы, сетевые интерфейсы и сетевые процессоры, имеет множество последовательных интерфейсов и различных сигналов, которые производят передачу данных и сигналов управления через многие домены синхрочастоты. Современные устройства становятся все более и более сложными, термин SoC, «система на кристалле», — теперь уже не новость, а реальность. Интеграция микросхем возрастает, растут тактовые частоты, и это возможно только при том, что сами размеры микросхем становятся все меньше. В свою очередь это приводит к тому, что многие ранее существовавшие параллельные интерфейсы постепенно заменяются на последовательные. Из этого следует, что современные микросхемы должны «уметь» работать с интерфейсами, имеющими различные тактовые частоты. Причем все эти частоты, в общем случае, могут быть не синхронными, одна относительно другой.

В этом разделе будут рассмотрены проблемы, возникающие у разработчика в том случае, если в проекте присутствуют несколько асинхронных тактовых частот.

Для того чтобы выполнить, а затем и проверить большой проект, его обычно делят на части. Если в проекте присутствует несколько асинхронных частот, то целесообразнее бывает делить проект на части так, чтобы в каждой из них действовала только одна тактовая частота. Такую часть проекта называют «домен синхрочастоты», или «клоковый домен». Симуляция такой части проекта не представляет особых трудностей. Трудности начинаются там, где необходимо пересечь границы такого домена. Когда сигнал из одного домена синхрочастоты приходит в другой домен, то этот сигнал в новом домене синхрочастоты будет асинхронным.

Для того чтобы разрабатывать специализированные интегральные схемы и выполнять проекты в FPGA, инженеры имеют много программных инструментов, которые могут компилировать проекты и трассировать схемы

## Краткий курс HDL. Часть 11. Асинхронные частоты, пересечение клоковых доменов и синхронизация

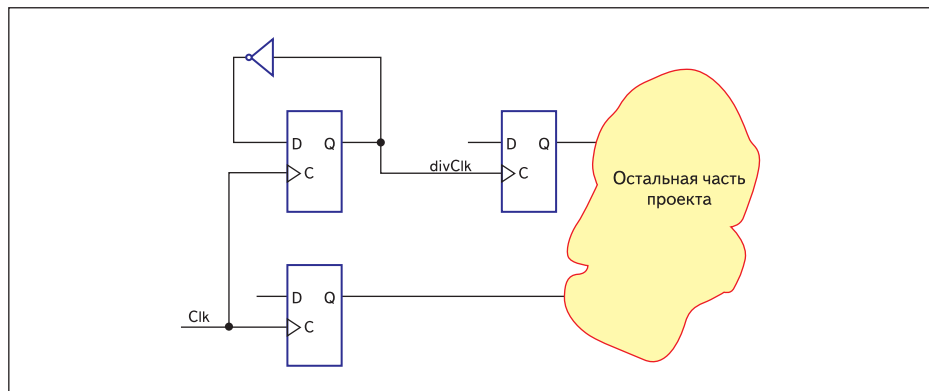


Рис. 1. Проект, имеющий единственный домен синхрочастоты

в миллионы вентилях, но эти программы не решают проблему синхронизации сигнала. Инженер-проектировщик должен знать надежные методики проектирования, которые уменьшают риск отказа для схем, в которых имеется несколько доменов синхрочастоты.

### Основы CDC

В этом разделе будут описаны проблемы, возникающие в проекте при пересечении клоковых доменов или, говоря по-другому, при пересечении домена синхрочастоты (CDC). Но сначала давайте рассмотрим некоторые основы CDC.

### Домен синхрочастоты

Домен синхрочастоты представляет собой ту часть проекта, которая тактируется одной или несколькими синхрочастотами, причем все эти синхрочастоты должны иметь постоянные сдвиги фазы. Если в какой-либо части проекта имеется синхрочастота или инвертированная синхрочастота, или синхрочастота, полученная из исходной путем деления на 2, то такая часть проекта считается клоковым доменом с одной синхрочастотой. Если же домены имеют синхрочастоты с переменной фазой и соотношениями времени, то такие домены считают доменами с различными синхрочастотами.

На рис. 1 показано, что проект имеет единственный домен синхрочастоты, потому что

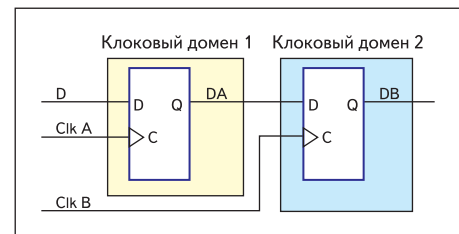


Рис. 2. Путь домена синхрочастоты

синхрочастота *divClk* получается из тактовой частоты от задающего генератора синхронизации *Clk*, путем деления ее на 2.

На рис. 2 показано несколько синхрочастот от различных источников. Ту часть проекта, которая управляется этими синхрочастотами, называют доменами синхрочастоты, и сигналы, осуществляющие передачу импульсов между этими асинхронными доменами синхрочастоты, называют путями пересечения домена синхрочастоты (CDC, clock domain crossing). Сигнал *DA* считают асинхронным сигналом в домене синхрочастоты, так как между генератором синхронизации *A* (*Clk A*) и генератором синхронизации *B* (*Clk B*) не существуют постоянные соотношения фазы и времени.

### Основные принципы

Первый шаг при разработке мультисинхрочастотных проектов состоит в том, что разработчик должен понять проблему стабильности сигнала. Когда сигнал пересекает до-

мен синхрочастоты, то он появляется в новом домене синхрочастоты как асинхронный сигнал. Схема, которая принимает этот сигнал, должна его засинхронизировать, то есть сделать его в этом новом домене стабильным. Синхронизация предотвращает в новом домене синхрочастоты метастабильное состояние первого запоминающего элемента схемы (триггера), и это позволяет в новом домене работать уже со стабильным сигналом.

Метастабильность — это неспособность триггера достигнуть известного состояния в определенный момент времени. Когда триггер входит в метастабильное состояние, то невозможно предсказать ни уровень выходного напряжения элемента, ни период времени, за который этот выход перейдет к правильному уровню напряжения. В течение этого переходного времени выход триггера будет находиться на некотором промежуточном уровне напряжения или колебаться и может передать этот недопустимый уровень сигнала со своего выхода к другим триггерам схемы.

### Основы асинхронных сигналов

Асинхронные сигналы, включая CDC-сигналы, могут быть метастабильными.

#### Метастабильность

Правильный режим работы триггера, имеющего импульсы синхронизации, зависит от того, насколько будет устойчив сигнал на входе данных относительно синхроимпульса. Причем нас интересует, чтобы входной сигнал данных для любого триггера был устойчивым для определенного промежутка времени до (setup —  $t_{SU}$ ) и после (hold —  $t_{HD}$ ) фронта синхроимпульса. Если эти требования по времени задержки будут соответствовать, то, соответственно, после максимальной задержки  $t_{CO}$  (clock-to-output delay — задержка выхода относительно синхрочастоты) правильный сигнал появится на выходах триггера, и этот сигнал будет иметь достоверный выходной уровень (VoL или VoH) в выходе триггера. Однако если интервалы времени по предустановке и интервалы времени по удержанию сигнала не соответствуют требуемым, то может потребоваться значительно больше времени, чем  $t_{CO}$ , для того, чтобы выход триггера смог достигнуть достоверного логического уровня. Это называют непостоянным поведением, или метастабильностью.

Итак, для записи данных в триггер необходимо соблюдать требование по минимальному времени предустановки данных и времени их удержания и только при соблюдении этих условий входные данные будут надежно синхронизированы. Но в действительности вход данных, поступающих на триггер D, является асинхронным по отношению к синхрочастоте. Поступление правильных входных данных можно проиллюстрировать на примере интервала времени относительно фронта тактовой частоты в **Окне выбора** (Decision

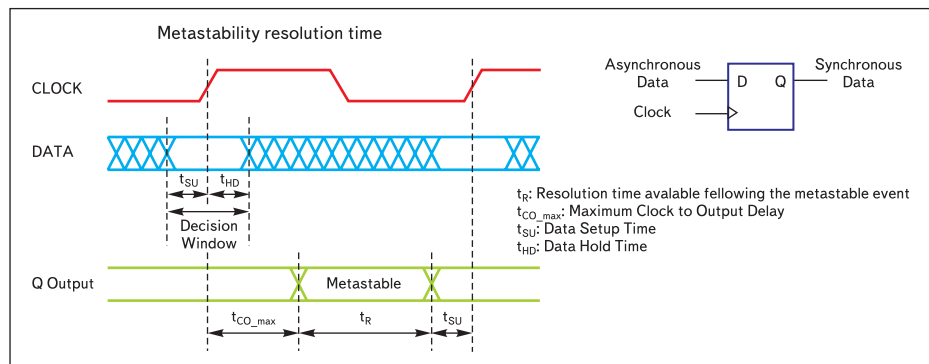


Рис. 3. Однокаскадный синхронизатор

Window), что показано на рис. 3. Если входные данные будут меняться вне **Окна выбора**, то триггер захватит правильные данные. Но поскольку время поступления входных данных относительно синхрочастоты неизвестно, то их изменение может произойти внутри **Окна выбора**, и в этом случае окно может рассматриваться как некоторая «опасная зона».

После задержки, определяемой параметром  $t_{CO}$ , входные данные появляются на выходе триггера — Q. Если входные данные попадут в «опасную зону», то выход Q, вероятно, будет в метастабильном состоянии до тех пор, пока триггер не перейдет в состояние логической единицы или нуля. Дополнительное время, требуемое для перехода триггера в установившееся логическое состояние, называют временем разрешающей способности — resolution time ( $t_R$ ). На рис. 4 показана зависимость ширины **Окна выбора** от параметра задержки  $t_{CO}$ . Такую диаграмму будем называть **Окном метастабильности** — Metastability Window (MW).

Промежутки времени setup и hold зависят от того, как именно в микросхеме выполнен триггер, от технологии его изготовления, эксплуатационных режимов и нагрузки на выходах, особенно на тех выходах, которые не буферизированы. Чем более крутые фронты имеют входные сигналы, и чем меньше нестабильность фронтов у этих сигналов, тем меньшее время необходимо для предустановки сигнала.

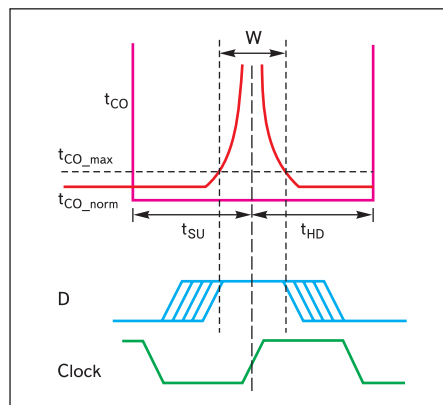


Рис. 4. Окно метастабильности

Изготовители FPGA и ИС нормируют параметры для своих триггеров. Для того чтобы определить вероятность отказа триггера, используется статистический параметр — среднее время безотказной работы (среднее время между отказами) (MTBF, Mean Time Between Failures), который и описывает характеристику метастабильности триггера. Изготовители рассчитывают среднее время безотказной работы в зависимости от ширины окна времени, в течение которого изменение во входном сигнале заставляет триггер становиться нестабильным. Кроме этого, для вычисления среднего времени безотказной работы используется частота входного сигнала и значение синхрочастоты, управляющей триггером.

Каждый тип триггера, описанный производителями в библиотеках для специализированных интегральных схем или FPGA, имеет свои требования по синхронизации. Это позволяет разработчику точно определить временные параметры, такие как время предустановки сигнала (Setup time) и время удержания сигнала (Hold time). Такие спецификации изготовителей учитывают множество параметров, чтобы объяснить все возможные влияния в зависимости от напряжения питания, рабочей температуры, качества входного сигнала и качества изготовления микросхем.

Если проект соответствует всем требованиям по синхронизации, то возможность сбоя для триггера будет минимальна. Программы синтеза в современных проектах по ИС и FPGA гарантируют, что в цифровых схемах, обработанных ими, все триггеры в этих проектах отвечают требованиям по предустановке и удержанию сигнала, но асинхронные сигналы создают проблемы для программного обеспечения. Сигнал, пересекающий домен синхрочастоты, будет асинхронным по отношению к логике, находящейся в новом домене синхрочастоты. Большинство программ синтеза не может определить, отвечают ли асинхронные сигналы требованиям синхронизации для триггеров. Поскольку они не могут определить параметры по времени, то, соответственно, и триггер для них является нестабильным, поэтому они не могут определить полную задержку в цепи,

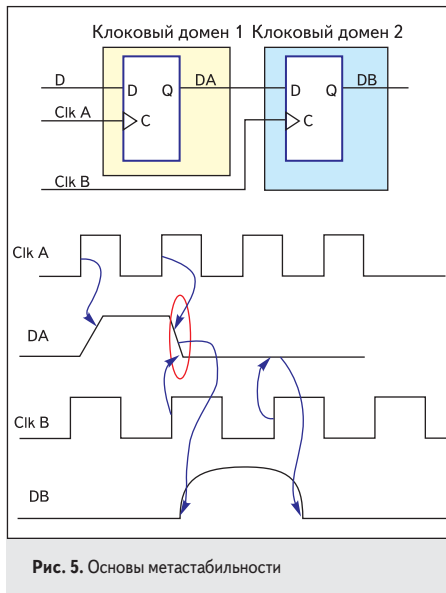


Рис. 5. Основы метастабильности

начинающейся от этого триггера и проходящей через комбинационную логику к следующему триггеру.

Продолжим рассмотрение метастабильности. Чем больше частота входного сигнала, подаваемого на триггер, тем больше риск того, что входной сигнал не будет соответствовать заданным параметрам по времени предустановки.

Как показано на рис. 5, если в схеме имеется генератор синхросигнала B, то частота синхронизации Clk B будет сэмплировать данные DA, поступающие с выхода первого триггера. Если выходной сигнал первого триггера DA изменится, например, фронт сигнала DA попадет на фронт синхросигнала Clk B, то сигнал на выходе второго триггера (DB) будет метастабильным.

Метастабильности нельзя избежать, но можно предпринять специальные решения для того, чтобы обрабатывать метастабильный сигнал. Лучший способ в этом случае — использовать схемы, которые устраняют проблемы с асинхронными сигналами, что позволит всему проекту работать правильно.

### Решение проблемы метастабильности

Два триггера, включенных последовательно, — это самое известное и самое легкое решение проблемы метастабильности. Рассмотрим это решение. Время разрешающей способности изменяется по экспоненте в соответствии с размером **Окна метастабильности**. Другими словами, если данные только чуть-чуть «не укладываются» в требования по времени на предустановку/удержание данных, то задержка, с которой триггер выдаст данные на выход будет только немного дольше, чем  $t_{CO}$ . Когда данные, поступающие на вход триггера, попадают в середину **Окна метастабильности**, то время задержки, с которой сработает триггер, будет намного большим, что показано на рис. 5 (изменение сигнала на выходе DB).

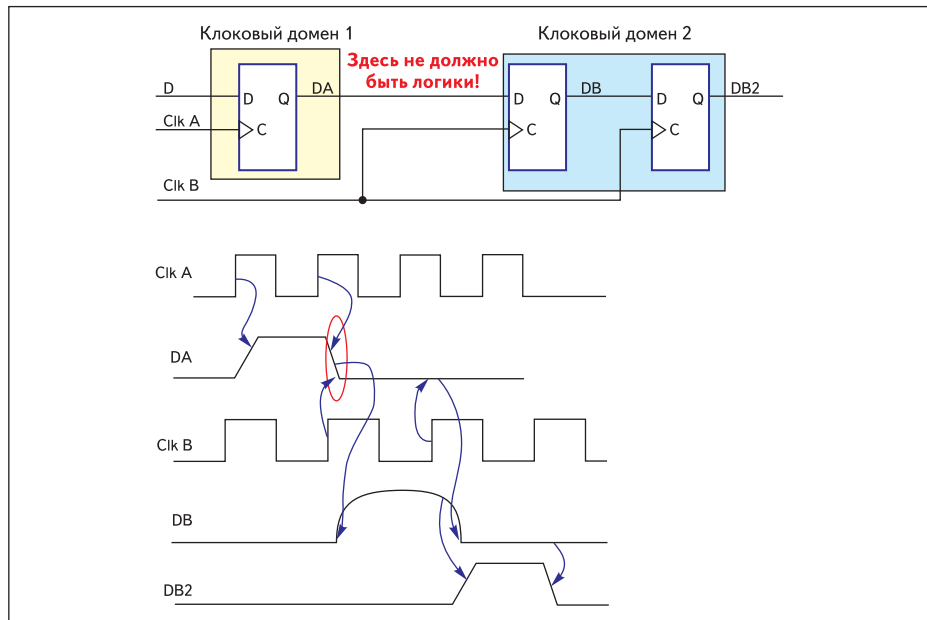


Рис. 6. Схема синхронизатора сигнала

Когда выработанный в первом триггере сигнал DB достигает второго триггера, то вероятность того, что переход сигнала попадет на вершину Окна метастабильности для входа данных второго триггера, очень низка. Поэтому практическая схема, состоящая из двух каскадно-включенных триггеров, значительно уменьшает вероятность отказа.

### Синхронизация сигнала

Сигналы синхронизируются для того, чтобы защитить всю логику в этом домене синхросигнала от метастабильного состояния триггера, на который приходит асинхронный, для этого домена, сигнал. Простейший синхронизатор представляет собой два триггера, включенных последовательно без какой-либо комбинационной схемы между ними. Такая схема проекта гарантирует, что первый триггер выходит из своего метастабильного состояния, и его выход переходит в устойчивое состояние перед тем, как второй триггер защелкивает это состояние. Необходимо также разместить эти триггеры, насколько возможно, ближе друг к другу для того, чтобы гарантировать наименьшую расфазировку тактовых сигналов между ними. Для этих целей фирмы изготовители ИС предлагают свои библиотечные компоненты — ячейки синхронизатора. Первый тип — это ячейки, которые обычно включают триггер с очень высоким усилением, хотя такой триггер и использует больше мощности и имеет размеры, которые больше, чем у стандартного триггера. Такой триггер имеет уменьшенное время на предустановку/удержание для входного сигнала и не создает колебательных процессов в схеме, когда входной сигнал вызывает метастабильность.

Другой тип ячейки синхронизатора представляет собой два близко расположенных

триггера без какой-либо комбинационной логики между ними. Для того чтобы синхронизация работала должным образом, сигнал, пересекающий домен синхросигнала, должен проследовать от триггера в домене синхросигнала — источника сигнала — к первому триггеру синхронизатора, не проходя через какую-либо комбинационную логику между этими двумя триггерами (рис. 6). В схеме синхронизатора сигнал, пересекающий домены синхросигнала, должен пройти от триггера, источника этого сигнала в первом домене синхросигнала, до первого триггера в синхронизаторе, не взаимодействуя при этом с какой-либо комбинационной логикой.

Это требование очень важно, потому что первый каскад синхронизатора чувствителен к сбоям, которые может производить комбинационная логика, например из-за гонки фронтов. Сбой, имеющий достаточную длительность (glitch) и происходящий в «правильное» время, может соответствовать требованиям по предустановке/удержанию для входного сигнала для первого триггера в синхронизаторе, а это приведет к тому, что синхронизатор передаст ложный импульс как достоверные данные для всей остальной части логики в новом домене синхросигнала. Синхронизированный сигнал будет достоверным в новом домене синхросигнала после задержки на два фронта синхроимпульса. Задержка для сигнала, приходящего в новый домен синхросигнала, по времени составит длительность от одного и до двух периодов синхросигнала в новом домене. Эмпирическое правило таково: схема синхронизатора создает два тактовых цикла задержки в новом домене синхросигнала, и проектировщик должен просто рассматривать это как дополнительную задержку в синхронизации сигналов, пересекающих домены синхросигнала. Есть много проектов для

Таблица 1. Виды синхронизаторов

Тип синхронизатора	Применение	Вход	Выход	Описание
Работающий по уровню	Определяет уровень сигнала	Уровень сигнала	Уровень сигнала	Вход должен быть достоверен в течение не менее 2 периодов синхрочастоты в новом домене. Каждый раз, когда выход синхронизатора переходит в активное состояние, вырабатывается только один признак — event
Работающий по фронту	Определяет положительный или отрицательный фронты сигнала	Уровень сигнала или импульс	Импульс сигнала	Вход должен быть достоверен в течение не менее 2 периодов синхрочастоты в новом домене
Работающий по приходящему импульсу	Синхронизируется по одному импульсу длительностью в один период синхрочастоты фронта сигнала	Импульс сигнала	Импульс сигнала	Входные импульсы должны иметь паузу не менее 2 периодов синхрочастоты в новом домене

синхронизаторов (табл. 1), потому что ни один из этих синхронизаторов не работает одинаково хорошо во всех приложениях.

Синхронизаторы относятся к одной из трех основных групп:

- работающие по уровню;
- работающие по фронту;
- работающие по приходящему импульсу.

Надо отметить, что хотя существуют и другие проекты синхронизаторов, но приведенные группы синхронизаторов могут быть использованы в большинстве приложений, с которыми сталкивается разработчик.

Вот в этом месте, скорее всего, читатель скажет: «По фронту, по уровню... Кажется, где-то это уже было...» А потом воскликнет: «Да ведь так работает любой контроллер прерываний, начиная от Intel-8259!» Конечно, и ведь входной сигнал запроса на прерывание — это как раз и есть тот самый случай, когда асинхронный сигнал попадает в новый синхронный домен, то есть в контроллер прерываний или в микроконтроллер, работающие на собственной синхрочастоте.

В синхронизаторе уровня сигнал, пересекающий домен синхрочастоты, должен оставаться высоким или низким для более чем двух тактовых циклов в новом домене синхрочастоты.

Требования, предъявляемые к этой схеме, состоят в том, что сигнал должен перейти в другое, неактивное состояние, прежде чем он может снова перейти в активное состояние. И такой переход будет трактоваться как новое достоверное событие. Например, в том случае, если мы считаем, что активное состояние для сигнала — это состояние высокого уровня, то для того, чтобы снова вызвать срабатывание синхронизатора, необходимо сначала сменить уровень сигнала на низкий, а потом снова подать активный уровень сигнала, то есть высокий уровень. И каждый раз после такой смены уровней синхронизатор будет считать переход в активное состояние событием (event). При этом синхронизатор может считать такое событие сколь угодно долгом. Эта схема представляет собой основу для всех других типов синхронизаторов. Однако есть случаи, когда нам необходим не уровень сигнала, а только его фронт. В схеме синхронизатора, работающей по фронту, используется схема синхронизатора по уровню как основа, к выходу которой добавляется еще один триггер (рис. 7).

Выход дополнительного триггера инвертирован и взят по «И» с выходом синхронизатора уровня. Эта схема обнаруживает положительный фронт входного сигнала для синхронизатора, он и генерирует широкий импульс, который будет активным в состоянии высокого уровня. Если вход синхронизатора пропустить через инвертор, то получим синхронизатор, который обнаруживает отрицательный фронт входного сигнала. Изменяя вентиль «И» на вентиль «И-НЕ», получим схему, которая генерирует активное состояние низким уровнем.

Если сигнал проходит из домена медленной синхрочастоты в домен с более быстрой синхрочастотой, то в таком случае синхронизатор, работающий по фронту, успеет обрабатывать не только фронт, но и импульс входящего сигнала.

Синхронизатор, работающий по фронту, выдает импульс в том случае, когда он обнаруживает положительный или отрицательный фронты входного сигнала. Но для этого синхронизатора присуще одно ограничение: длительность входного импульса должна быть больше, чем период синхрочастоты син-

хронизатора плюс необходимое время задержки его первого триггера. Самая безопасная длительность импульса в этом случае — это два периода синхрочастоты синхронизатора. Но такой синхронизатор не работает, если ему на вход подается импульс, длительность которого равна одному периоду синхрочастоты, и этот сигнал приходит в домен с более медленной синхрочастотой. В таком случае синхронизатор, работающий по приходящему импульсу, решает эту проблему. Входной сигнал для такого синхронизатора представляет собой один импульс, длительность которого равна периоду его синхрочастоты, и этот импульс заставляет схему синхронизатора сделать необходимые переключения в своем домене синхрочастоты (рис. 8). По входному импульсу с длительностью в один такт синхрочастоты первого домена формируется один выходной импульс с длительностью в один такт синхрочастоты второго домена.

Выход схемы переключается от высокого уровня к низкому и снова возвращается в высокий уровень каждый раз, когда импульс проходит через синхронизатор уровня. Для приходящего импульса длительностью в один тактовый цикл каждый раз, когда цепь, определяющая переключение сигнала, находит переключение и производит изменение состояния схемы, выход такого синхронизатора генерирует единственный импульс, длительность которого равна периоду одного тактового цикла синхрочастоты. Основная функция синхронизатора импульса состоит в том, чтобы, получив один входной импульс с длительностью в один такт синхрочастоты

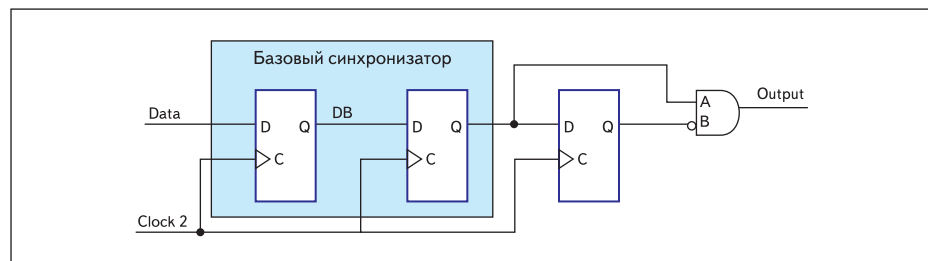


Рис. 7. Синхронизатор, работающей по фронту, состоит из базового синхронизатора, дополнительного триггера и инвертора

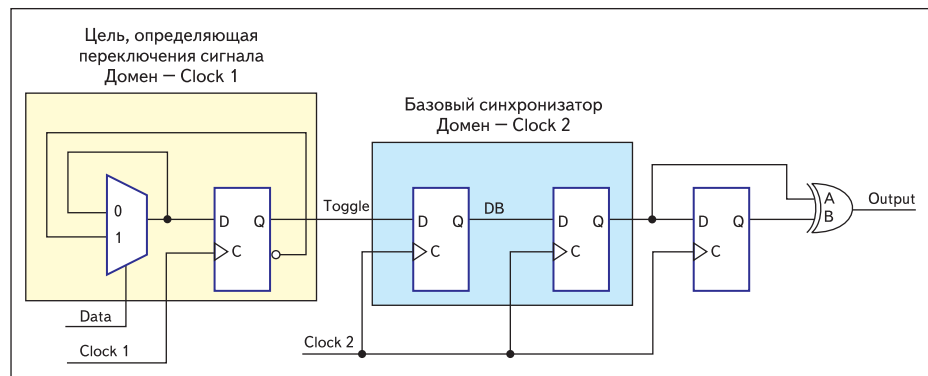


Рис. 8. Синхронизатор, работающий по приходящему импульсу

первого домена, сформировать один выходной импульс с длительностью в один такт синхрочастоты второго домена. У этого синхронизатора также есть ограничение для обрабатываемого импульса: между входными импульсами должен быть минимальный интервал, равный двум периодам синхрочастоты синхронизатора. Если входные импульсы будут расположены друг к другу ближе, чем выходные импульсы в новом домене синхрочастоты, то на выходе импульсы перекроются, и суммарный выходной импульс получится шире, чем один тактовый цикл. Эта проблема довольно серьезна, когда период входного импульса более чем в два раза длиннее, чем период синхрочастоты синхронизатора. В том случае, когда входные импульсы следуют слишком часто, синхронизатор не успевает обнаружить каждый импульс.

### Процедура установления связи (Handshaking) и FIFO

Во многих проектах необходимо передавать из одного домена в другой не только несколько одиночных сигналов, но и сигналы типа шин: шины данных, адреса и шины управления.

В этих задачах активно используются буферы типа FIFO и специальные протоколы для процедуры установления связи. Такие протоколы в отечественной аппаратуре называются «запрос–подтверждение», а в английском — Handshaking. Процедура установления связи позволяет цифровым схемам эффективно общаться друг с другом в том случае, когда время ответа одной или обеих схем непредсказуемо. Например, арбитр шины позволяет нескольким схемам просить разрешение на доступ к шине с разделяемыми ресурсами. Примерами таких шин могут быть шины PCI или AMBA (расширенная архитектура шины микроконтроллера). Блок арбитража производит проверку запросов к шине и определяет, кто получит к ней доступ. Каждая из схем, которая нуждается в доступе к шине, выставляет свой запрос на доступ, а арбитражная логика определяет, какой из запросов будет иметь наивысший

приоритет. Арбитр шины выставит сигнал подтверждения на доступ к шине той схеме, которая выставила сигнал запроса с наивысшим приоритетом. Получив подтверждение запроса, схема снимет сигнал запроса и начнет транзакцию шины.

Для работы схем, находящихся в различных доменах синхрочастоты, используют два базовых типа протокола установления связи: полный протокол установления связи и неполный протокол (Full- and Partial-Handshake Signaling). Каждый тип установления связи использует синхронизаторы, и каждый имеет свои варианты выполнения проекта. В случае полного протокола установления связи обе схемы ждут друг друга, когда будет выставлен или снят соответствующий сигнал установления связи. В качестве всем известного полного аппаратного протокола может быть приведен режим работы СОМ-порта с сигналами DTR, DSR. Также можно в качестве примера привести любой протокол асинхронной процессорной шины, имеющей сигнал «готовность».

Работа выполняется в два этапа. На первом этапе идет обмен сигналами, который приводит к выставлению активных состояний для сигналов, участвующих в обмене. При этом в той фазе, когда есть и сигнал запроса, и сигнал подтверждения, может производиться обмен данными или какие-либо другие действия по передаче данных между доменами. На втором этапе производится переход сигналов синхронизации из активного состояния в неактивное. Давайте рассмотрим это более подробно (рис. 9).

Сначала схема А устанавливает свой сигнал запроса. При этом схема В проверяет, что сигнал запроса является достоверным, и если находит, что пришел сигнал запроса от схемы А, то схема В в ответ на пришедший запрос устанавливает свой сигнал подтверждения. Схема А, выставив свой сигнал запроса, начинает проверять наличие сигнала подтверждения, который к ней должен прийти от схемы А. И если сигнал подтверждения к схеме А уже пришел, то она снимает свой сигнал запроса. В ответ на это схема В, определив, что запрос снят, снимает и свой сиг-

нал подтверждения. Схема А не выдает новый запрос до тех пор, пока она не обнаруживает, что сигнал подтверждения снят. Этот тип установления связи использует синхронизаторы, работающие по уровню. Данную методику можно применить в том случае, когда подтверждающая схема (В) должна сообщить схеме, выставившей запрос (А), что она активно обрабатывает текущий запрос. Это установление связи требует, чтобы схема, затребовавшая запрос, не выставляла свой следующий запрос до тех пор, пока она не получит сигнал подтверждения запроса и не сбросит свой текущий запрос.

Чтобы определять время, необходимое для обработки синхронизации этого типа протокола, используйте следующее эмпирическое правило: для того, чтобы сигналы пересекли домен синхрочастоты, необходимо потратить два тактовых цикла на синхронизацию, и при этом сигналы, прежде чем они пересекнут домен синхрочастоты, должны быть защелкнуты в регистрах. Весь обмен сигналами имеет максимальную длительность: пять циклов синхрочастоты домена А плюс максимум шесть циклов синхрочастоты в домене В. Полная процедура установления связи устойчива, потому что каждая схема явно знает состояние другой, проверяя сигналы запроса и подтверждения. Недостаток этой схемы состоит в том, что весь процесс использует много тактовых циклов для завершения транзакции.

Частичная процедура установления связи — это другая методика для обмена сигналами, в которой произведено сокращение описанной последовательности событий. При частичном установлении связи обе схемы, обменивающиеся друг с другом сигналами, уже не ждут того момента, когда партнер по обмену сигналами переведет свои сигналы в неактивное состояние. Частичная процедура установления связи менее устойчива, чем полная, потому что сигналы установления связи не указывают состояние обеих схем, и каждая схема должна сохранять информацию о состояниях, которые обычно присутствуют в установлении связи с полным набором сигналов. Но поскольку каждая из схем, участвующих

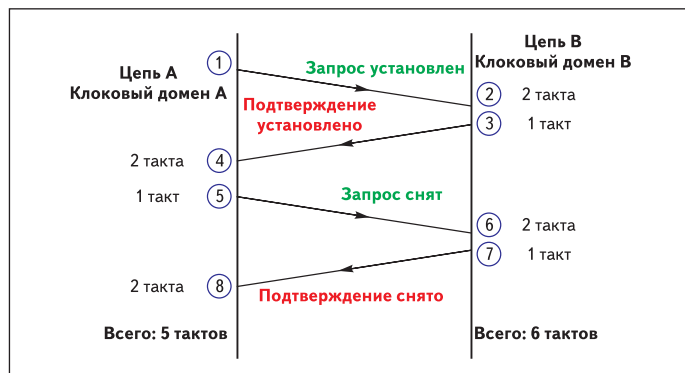


Рис. 9. Полный протокол установления связи (две схемы ждут друг друга перед тем, как установить или сбросить свои сигналы установления связи)

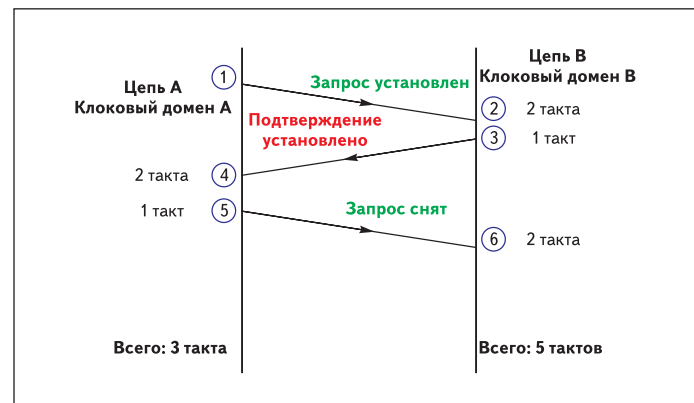


Рис. 10. Неполный протокол установления связи

в обмене сигналами, уже не ждет, пока другая схема снимет свои сигналы установления связи, то общая длительность процедуры обмена занимает меньше времени.

В том случае, когда используется протокол частичного установления связи, схема подтверждения должна генерировать свой сигнал за определенное время. Если схема подтверждения должна закончить обработку запроса прежде, чем его начнет обрабатывать другой узел схемы, то в этом случае время выставления сигнала подтверждения очень важно. Схема использует свой сигнал подтверждения, чтобы сообщить о том моменте времени, когда она закончила обработку.

В качестве примера здесь можно привести работу шины PCI. Абоненты шины должны выставить свой сигнал подтверждения за время от одного до трех периодов синхрочастоты шины. В четвертом же периоде сигнал подтверждения выставит мост, связывающий шины.

Одни схемы частичного установления связи выполняются так, чтобы они работали и по уровню, и по импульсу, а другие схемы работают только по импульсу. В первой схеме частичного установления связи схема А устанавливает свой сигнал запроса в активный уровень, и схема В подтверждает приход запроса своим импульсом, имеющим длительность в один период синхрочастоты. В этом случае схема В не проверяет, когда схема А снимет свой сигнал запроса. Однако, чтобы работать по этому методу, схема А должна снимать свой сигнал запроса за время одного тактового цикла, иначе схема В не может различить предыдущий запрос и новый запрос. При таком установлении связи схема В использует синхронизатор, работающий по уровню для сигнала запроса, а схема А для обработки сигнала подтверждения использует синхронизатор, работающий по импульсу. Импульсы подтверждения вырабатываются только тогда, когда схема В обнаруживает сигнал запроса. Эта ситуация позволяет схеме А управлять интервалом следования импульсов, которые она получает на входе синхронизатора, управляя своим сигналом запроса (рис. 10). Схема А устанавливает свой запрос, а схема В подтверждает его своим импульсом, длительность которого равна периоду тактового импульса. Две схемы ждут друг друга перед тем, как установить или сбросить свои сигналы установления связи.

Итак, еще раз необходимо отметить: для того чтобы определить задержки по времени при синхронизации, используйте следующее эмпирическое правило. Сигналам для того, чтобы пересечь домен синхрочастоты, необходимо два тактовых цикла, и прежде чем сигналы будут пересекать домен синхрочастоты, эти сигналы должны быть защелкнуты в регистре.

Весь обмен сигналами имеет максимальную длительность: три цикла синхрочастоты домена А плюс максимум пять циклов синхро-

Таблица 2. Виды синхронизаторов

Тип синхронизатора	Схема	Тип сигнала	Длительность обработки	Синхронизатор	Ограничения
Полный	Схема А (запрос)	Уровень сигнала	5 тактов синхрочастоты	По уровню сигнала	Последовательность обработки очень длительная. Запрос должен быть снят на стороне В как минимум на 2 периода синхрочастоты В
	Схема В (подтверждение)	Уровень сигнала	6 тактов синхрочастоты	По уровню сигнала	Подтверждение должно быть снято на стороне А как минимум на 2 периода синхрочастоты А.
Частичный 1	Схема А (запрос)	Уровень сигнала	3 такта синхрочастоты	По импульсу или по уровню сигнала	Надо контролировать скорость поступления импульсов подтверждения.
	Схема В (подтверждение)	Импульс сигнала	5 тактов синхрочастоты	По уровню сигнала	Запрос должен быть снят на стороне В как минимум на 2 периода синхрочастоты В
Частичный 2	Схема А (запрос)	Импульс сигнала	2 такта синхрочастоты	По импульсу или по уровню сигнала	Надо сохранять информацию о текущем запросе
	Схема В (подтверждение)	Импульс сигнала	3 такта синхрочастоты	По импульсу или по уровню сигнала	Надо защелкивать в регистрах информацию о запросе и о подтверждении

частоты в домене В. Процедура частичного установления связи имеет длительность на два тактовых цикла меньше в домене синхрочастоты А и на один тактовый цикл меньше в домене синхрочастоты В, чем диаграмма при полном установлении связи. Можно сократить диаграмму обмена еще на несколько тактовых циклов, используя другую схему частичного установления связи, в которой схема А устанавливает свой запрос уже не уровнем, а импульсом, имеющим длительность в один период синхрочастоты. Схема В получает этот запрос и также подтверждает его своим импульсом, имеющим длительность в один период синхрочастоты. В этом случае обе схемы должны сохранять свое состояние, чтобы «понимать», когда производится обработка запроса. В этом типе схем для установления связи используются синхронизаторы, работающие по импульсу. Но если есть возможность подать на схему синхрочастоту, которая будет вдвое выше основной синхрочастоты, то можно будет вместо синхронизатора, работающего по импульсу, использовать синхронизатор, работающий по фронтам (рис. 11). Схема А устанавливает свой запрос, а схема В подтверждает его своим импульсом, длительность которого равна периоду тактового импульса. Но если одна из схем имеет тактовую частоту вдвое выше, чем другая, и, соответственно, ее период тактовой частоты вдвое меньше, чем длительность приходящего импульса, то в этом варианте можно использовать синхронизатор, работающий по фронту.

Полная последовательность имеет длительность максимум два цикла в домене синхрочастоты А плюс максимум три цикла в домене синхрочастоты В. Эта методика процедуры частичного установления связи использует

на три тактовых цикла в домене синхрочастоты А меньше, чем процедура для полного установления связи, и еще на три тактовых цикла меньше в домене синхрочастоты В. Эта методика также работает быстрее, чем методика, приведенная в первом варианте для частичного установления связи. Она быстрее на один цикл в домене синхрочастоты А и на два цикла быстрее в домене синхрочастоты В (табл. 2). Эти протоколы установления связи задействуют в обмене только одиночные сигналы для пересечения доменов синхрочастоты. Однако когда домены синхрочастоты пересекать группы сигналов, необходимо использовать более сложные схемы. ■

## Литература

- Cummings. C. E. Synthesis and Scripting Techniques for Designing Multi-Asynchronous Clock Designs. Synopsys Users Group Conference, San Jose, CA, [www.sunburst-esign.com/papers](http://www.sunburst-esign.com/papers), March 2001.
- Clock Domain Crossing. Closing the Loop on Clock Domain Functional Implementation Problems. Technical paper. Cadence Design Systems, Inc. [www.cadence.com](http://www.cadence.com)
- Metastability in Lattice Devices. Technical Note TN1055. July 2007.
- Metastability in Altera Devices, ver. 4. Application Note 42. May 1999.
- Metastability Considerations. Application Note (Version 1.0). XAPP077. January, 1997.
- Metastability Characterization Report for Actel Flash FPGAs. <http://www.actel.com>
- Johnson H. W., Graham M. High Speed Digital Design (A Handbook of Black Magic). Prentice-Hall, 1993.
- Stein M. Crossing the abyss: asynchronous signals in a synchronous world. Paradigm Works. July 24, 2003. [www.edn.com](http://www.edn.com)

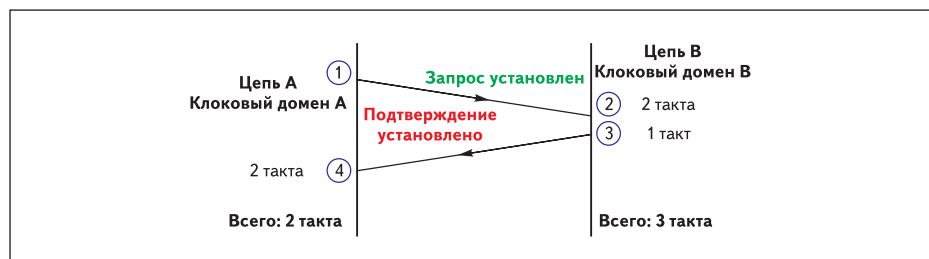


Рис. 11. Неполный протокол установления связи (в этом варианте используются синхронизаторы, работающие по приходящим импульсам)