

---

## Аппаратные средства

PLL в чипах семейства Cyclone III поддерживают несколько средств широкого применения для управления тактовыми сигналами. В этом разделе обсуждается реализация умножения и деления тактового сигнала, реализация фазового сдвига и программирования рабочих циклов.

### Умножение и деление тактового сигнала

Каждая PLL в семействе Cyclone III позволяет синтезировать тактовые сигналы для выходных портов PLL, используя коэффициенты масштабирования  $M/(N \times \text{пост-масштабирующий счётчик})$ . Входной тактовый сигнал делится на число  $N$ , а затем умножается на число  $M$  в схеме обратной связи. Обратная связь управляет VCO (ГУН), чтобы создать частоту  $f_{IN}$  ( $M/N$ ). Каждый выходной порт имеет свой собственный пост-масштабирующий счётчик, который делит высокую частоту VCO. Для выходов PLL с кратными частотами, значение частоты VCO задаётся как наименьшее общее кратное выходных частот. Например, если требуется выходная частота от одной из PLL на 33 и 66 МГц, программа Quartus II задаёт VCO частоту 660 МГц (наименьшее общее кратное частот 33 и 66 МГц для диапазона VCO). Затем на пост-масштабирующих счётчиках эта частота уменьшается для каждого выходного порта.

Для каждой PLL имеется один пред-масштабирующий счётчик  $N$  и один пост-масштабирующий счётчик  $M$ , каждый имеет диапазон от 1 до 512. Счётчик  $N$  не может быть использован для контроля над рабочими циклами, поскольку его назначение - только деление частоты. Для каждой PLL существуют пять обычных пост-масштабирующих счётчиков, которые управляют тактовыми сигналами для сетей GCLK или для внешних выходов. Эти пост-масштабирующие счётчики могут задавать диапазон от 1 до 512 с 50% рабочим циклом. Эти же пост-масштабирующие счётчики с диапазоном от 1 до 256 могут работать с настройкой рабочего цикла не 50%. Суммарное значение повышающих/понижающих счётчиков задаётся в проекте выбором значения делителя для выбранного счётчика.

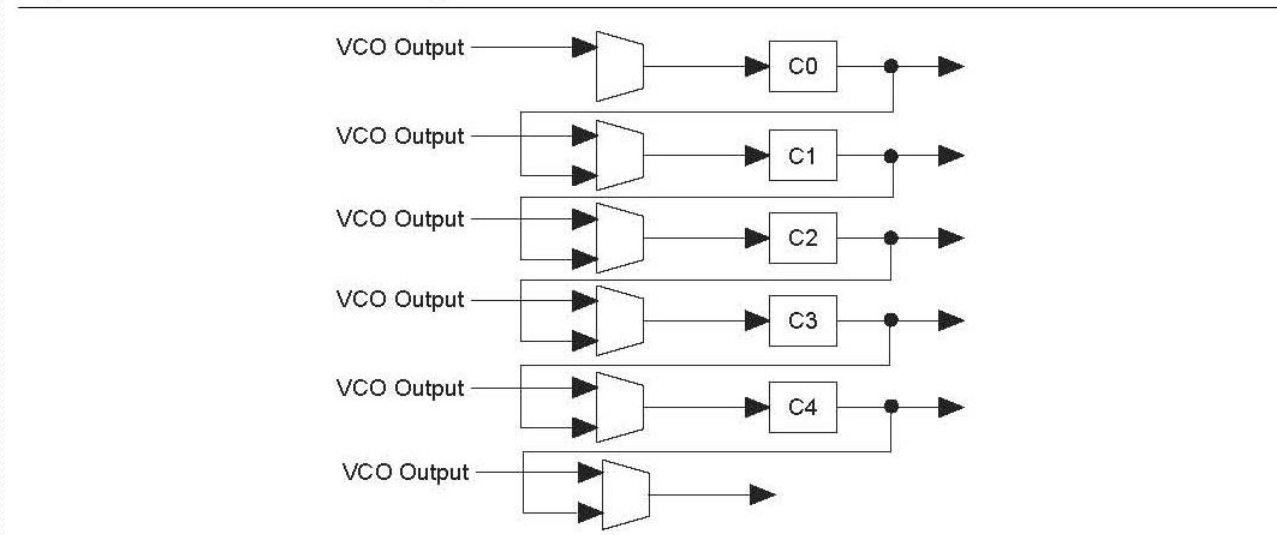
Программа Quartus II автоматически задаёт соответствующие параметры масштабирования в соответствии со значениями входной частоты, умножения и деления, введёнными в мегафункции ALTPLL.

Выравнивание по фазе между выходными счётчиками определяется с помощью спецификации  $t_{PLL\_PSERR}$ .

### Каскадирование пост-масштабирующих счётчиков

PLL семейства Cyclone III поддерживают каскадирование пост-масштабирующих счётчиков для создания счётчиков с коэффициентом более 512. Это реализуется путём подачи выхода с одного из С счётчиков на вход другого С счётчика, как показано на рис. 5-12.

**Figure 5–12. Counter Cascading**



Когда каскадируются счётчики для реализации большего деления высокочастотного тактового сигнала от VCO (ГУН), то эти счётчики рассматриваются как один счётчик с индивидуальными настройками счёта.

Например, если  $C0 = 4$  и  $C1 = 2$ , то каскадируемое значение будет  $C0 \times C1 = 8$ .

Каскадирование пост-масштабирующих счётчиков задаётся автоматически программой Quartus II в конфигурационном файле. Пост-масштабирующие счётчики не могут быть изменены при реконфигурации PLL.

### Программирование рабочего цикла

Программирование рабочего цикла позволяет генерировать на выходах PLL различные рабочие циклы. Это средство поддерживается на пост-масштабирующих счётчиках PLL. Вы можете изменять рабочие циклы путём уменьшения и увеличения настройки времени счёта для пост-масштабирующих счётчиков. Программа Quartus II использует входную частоту и необходимое значение умножения или деления для задания выбора рабочего цикла. Значение пост-масштабирующего счётчика определяет точность рабочего цикла. Точность является 50% от значения пост-масштабирующего счётчика. Например, если счётчик C0 равен 10, то точность будет 5%, что даст возможность выбрать рабочий цикл от 5 до 90% шагом в 5%.

Комбинирование программируемых рабочих циклов с программированием фазового сдвига позволяет генерировать прецизионные не перекрывающиеся тактовые сигналы.

### Контрольные сигналы PLL

Вы можете использовать три следующих сигнала для наблюдения и контроля над работой и ресинхронизацией PLL.

#### *pfdena*

Используйте сигнал *pfdena* для удержания последней защёлкнутой частоты, чтобы ваша система имела время для сохранения текущих настроек перед выключением. Сигнал *pfdena* контролирует выход PFD с помощью программируемого вентиля. Если вы запрещаете PFD, то VCO (ГУН) работает на последнем заданном значении, контролируя напряжение и частоту с некоторым длительным смещением в сторону снижения частоты.

#### *areset*

Сигнал *areset* - это сброс или ресинхронизация входа для каждой PLL. Входные выводы микросхемы или внутренняя логика могут управлять этим входным сигналом. Когда он установлен, сбрасываются счётчики PLL, выходы PLL, а сама PLL выводится из защёлки. VCO возвращаются его номинальные значения. Когда сигнал возвращается в сброшенное состояние, PLL ресинхронизируется по её выходам и перезащёлкивается.

Вы должны включать сигнал *areset* в ваш проект в одном из следующих случаев:

- когда в вашем проекте используется реконфигурация PLL или переключение тактовых сигналов,
- когда фазовая зависимость между входными и выходными тактовыми сигналами должна сохраняться после состояния потери защёлки.

Если входной тактовый сигнал PLL может подключаться или может быть нестабилен при подаче питания, используйте сигнал *areset* после того, как входной тактовый сигнал стабилизируется и будет соответствовать заданным параметрам.

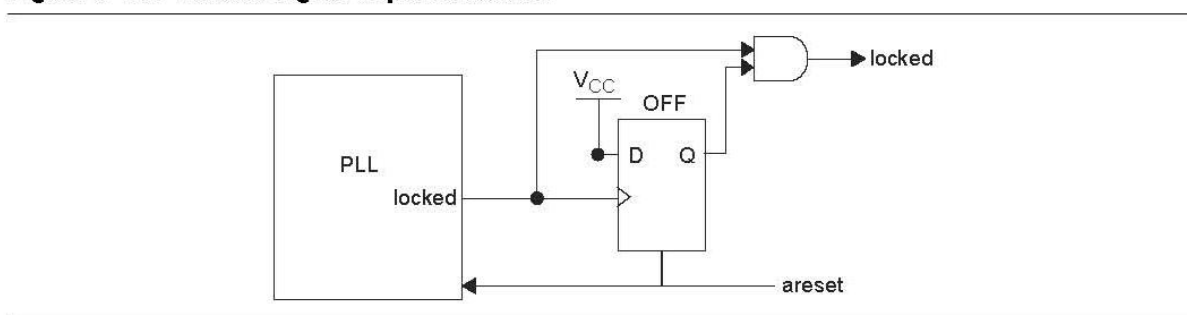
#### *locked*

Выход *locked* показывает, когда PLL защёлкивается по соответствующему тактовому сигналу, а выходы PLL работают на заданной в Quartus II MegaWizard™ Plug-in Manager частоте и фазе.

Altera рекомендует вам использовать сигналы *areset* и *locked* в вашем проекте для контроля и наблюдения за статусом вашей PLL.

Эта реализация показана на рис. 5-13.

**Figure 5–13. Locked Signal Implementation**



Если вы используете инструмент SignalTap® II, чтобы установить пробник на сигнал locked перед D-триггером, сигнал locked переходит в нуль, когда снимается сигнал areset. Если сигнал areset не разрешён, то специальная логика не реализуется в мегафункции ALTPLL.

За дополнительной информацией о контрольных сигналах PLL обратитесь к [руководству пользователя мегафункцией ALTPLL](#).

### Переключатель тактов

Средство переключателя тактов позволяет PLL переключаться между двумя поступающими тактовыми сигналами. Используйте это средство для резервирования тактового сигнала или для приложений с двумя тактовыми сигналами в одном домене, такими как система, которая переключается на запасной тактовый сигнал, когда предыдущий тактовый сигнал останавливается. Ваш проект может автоматически переключать тактовый сигнал, который долго не поступает, или переключать по сигналу пользователя clkswitch.

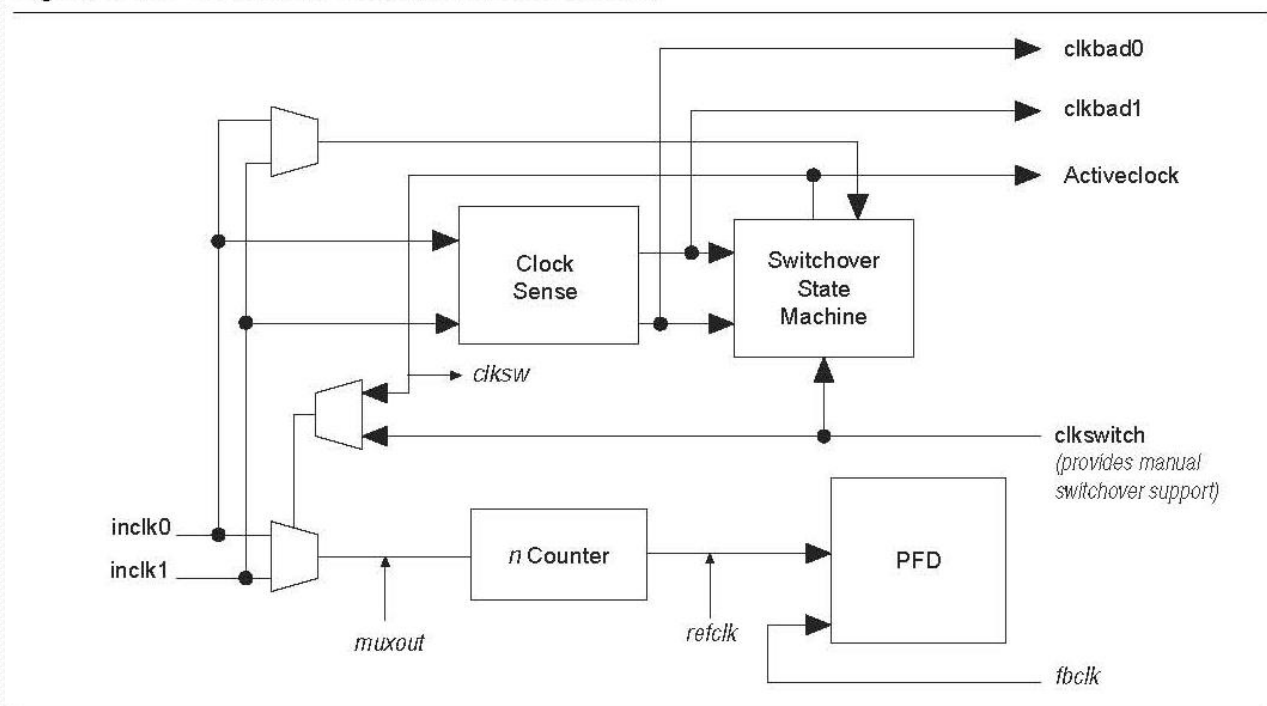
### Автоматический переключатель тактового сигнала

PLL семейства Cyclone III поддерживает полностью конфигурируемый переключатель тактового сигнала.

Когда текущий поступающий тактовый сигнал пропадает, блок чувствительности тактового сигнала автоматически переключается на запасной тактовый сигнал PLL. Схема переключателя тактового сигнала выдаёт три статусных сигнала - clkbad[0], clkbad[1] и activeclock - из PLL для реализации собственной схемы переключателя тактов. Вы можете выбрать источник тактового сигнала в качестве запасного тактового сигнала, подключив его к порту inclk1 PLL вашего проекта.

На рис. 5-14 показана блок-схема схемы переключателя тактового сигнала, реализованная в PLL.

**Figure 5-14. Automatic Clock Switchover Circuit**

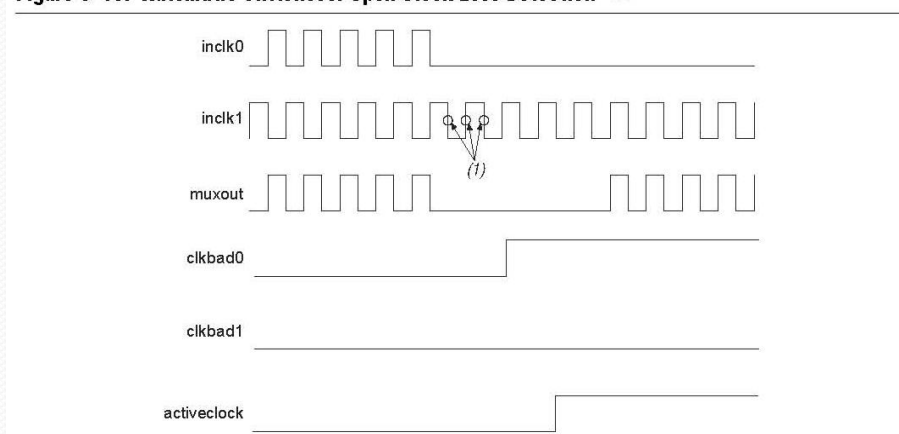


Существуют два способа использовать средство переключения тактов:

- Используйте схему переключателя для переключения входов `inclk0` на `inclk1`, тактируемых одинаковыми частотами. Например, в приложениях, где требуется запасной тактовый сигнал такой же частоты, что и основной, конечный автомат переключателя тактов генерирует сигнал, контролирующий вход выбора мультиплексора, показанного на рис. 5-14. В этом случае, `inclk1` становится основным тактовым сигналом PLL. Автоматический переключатель тактового сигнала может переключать входы `inclk0` и `inclk1` взад и вперёд любое количество раз, когда один из тактовых сигналов теряется, а другой является доступным в этот момент.
- Используйте вход `clkswitch` для управления состоянием через пользовательские или системные функции. Такая возможность одинаково применима для переключателя одинаковых или переключателя разных частот. Например, если `inclk0` имеет частоту 66 МГц, а `inclk1` - 200 МГц, вы должны контролировать переключатель, поскольку схема блока чувствительности не может отслеживать частоты первого и второго тактовых сигналов для частот, отличающихся более чем на 20 %. Это средство идеально подходит для случаев, когда источник тактового сигнала может происходить от нескольких подключаемых плат, и требуется системный контроль переключателя тактового сигнала между рабочими частотами. Выбирайте второй тактовый сигнал, чтобы VCO работал в рекомендованном диапазоне частот. Задайте соответствующие значения счётчиков M, N и C, чтобы сохранить рекомендованный частотный диапазон VCO.

На рис. 5-15 показаны временные диаграммы средства переключателя тактового сигнала, когда происходит автоматическое детектирование потери тактового сигнала. Здесь сигнал `inclk0` переходит в нуль. После того, как сигнал остаётся в нуле около двух тактовых циклов, схема блока чувствительности устанавливает в единицу сигнал `clkbld[0]`. Так как основной тактовый сигнал не поступает, конечный автомат переключателя управляет мультиплексором через сигнал `clksw`, чтобы переключиться на `inclk1`.

Figure 5-15. Automatic Switchover Upon Clock Loss Detection (1)



**Примечание к рис. 5-15:**

- (1) Переключатель разрешён на спадающем фронте `inclk0` или `inclk1`, в зависимости от того, какой из тактовых сигналов доступен. На этом рисунке переключатель доступен на спадающем фронте `inclk1`.



**Ручное управление**

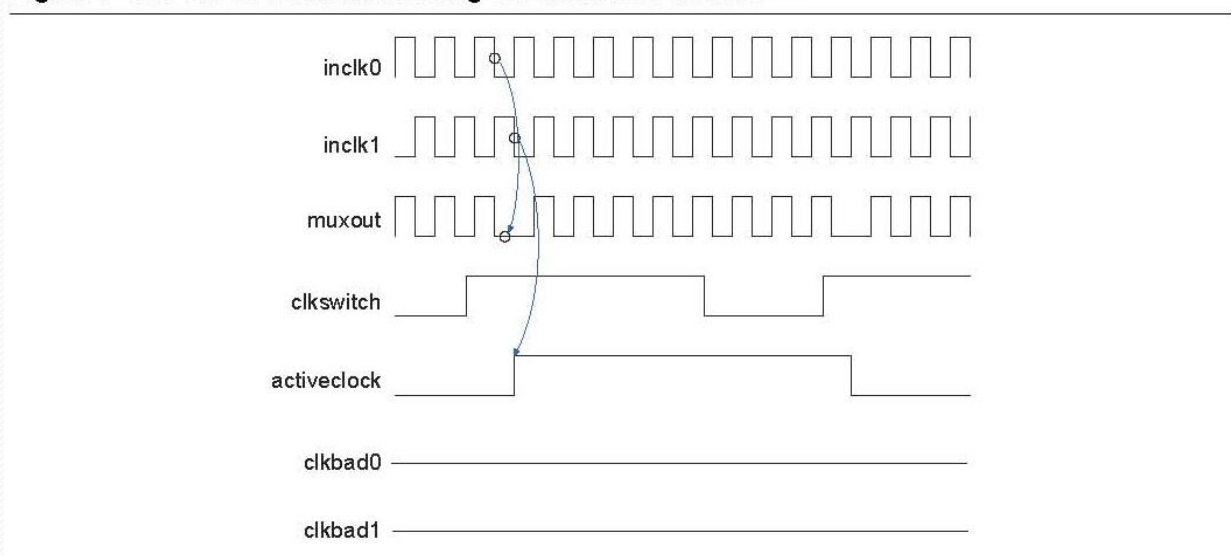
Если вы используете автоматический переключатель, вы должны переключать входы тактовых сигналов с помощью средства ручного управления по входу clkswitch.

На рис. 5-16 показан пример временных диаграмм, иллюстрирующих средство переключения тактов, когда оно контролируется clkswitch. В этом случае, оба источника тактов работают, а inclk0 используется в качестве основного тактового сигнала. По переходу из нуля в единицу сигнала clkswitch запускается операция переключения. Сигнал clkswitch должен оставаться в единице не менее трёх тактовых циклов (или не менее трёх наибольших тактовых периодов, если на входах inclk0 и inclk1 поступают разные частоты). По фронту спада сигнала inclk0, основной тактовый сигнал счётчика, muxout, закрывается, чтобы избежать любых выбросов тактового сигнала. По фронту спада сигнала на inclk1 мультиплексор основного тактового сигнала переключает входы PLL с inclk0 на inclk1, и сигнал activeclock меняет своё значение для отображения текущего основного тактового сигнала PLL.

В этом режиме сигнал activeclock дублирует сигнал clkswitch. Оба блока продолжают работу во время ручного переключения, при этом не появляется сигналов clkbad. Поскольку схема переключения реагирует на положительный фронт, фронт спада сигнала clkswitch не вызывает обратного переключения с входа inclk1 на вход inclk0. Когда сигнал clkswitch снова переходит в единицу, процесс повторяется. Сигнал clkswitch и автоматическое переключение работают только в зависимости от доступности тактовых сигналов, которые можно переключить. Если тактовые сигналы не доступны, конечный автомат ожидает момента, пока тактовые сигналы станут доступны.

Если CLKSWITCH = 1, функция автоматического переключателя отменяется. Пока сигнал clkswitch в единице, блокируется дальнейшее переключение тактового сигнала.

**Figure 5–16. Clock Switchover Using the clkswitch Control (1)**

**Примечание к рис. 5-16:**

- (1) Оба входа `inclk0` и `inclk1` должны быть тактируемы, когда сигнал `clkswitch` запускает процесс ручного переключения тактового сигнала.

**Ручной переключатель тактового сигнала**

PLL семейства Cyclone III поддерживают ручной переключатель тактового сигнала, при котором сигнал `clkswitch` контролирует, который из входов `inclk0` или `inclk1` является входом тактового сигнала PLL. Характеристики ручного переключателя тактового сигнала похожи на средство ручного управления в автоматическом переключателе тактового сигнала со схемой чувствительности к фронту. Когда сигнал `clkswitch` переходит в единицу, запускается переключатель. Фронт спада сигнала `clkswitch` не вызывает обратного переключения на предыдущий входной тактовый сигнал.

За дополнительной информацией о программной поддержке PLL в Quartus II обратитесь к [руководству пользователя мегафункцией ALTPLL](#).

**Рекомендации**

Используйте следующие рекомендации в ваших проектах с переключателем тактового сигнала PLL:

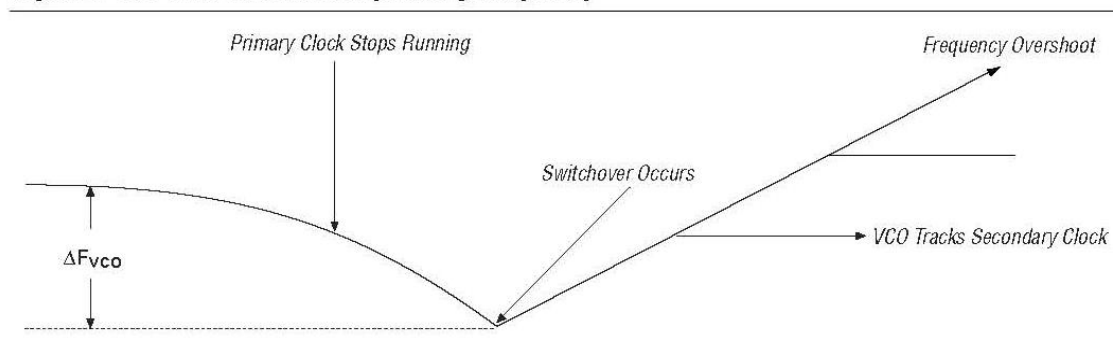
- Детектор потери тактового сигнала и автоматический переключатель тактов требуют, чтобы частоты `inclk0` и `inclk1` отличались бы друг от друга в пределах 20%. Не выполнение этого требования вызывает ошибочное появление сигналов `clkbad[0]` и `clkbad[1]`.
- При использовании ручного переключателя тактового сигнала, разница между `inclk0` и `inclk1` может быть более 20%. Однако, отличия между двумя источниками тактового сигнала (по частоте и(или) фазе) могут вызвать потерю защёлки PLL. Сбрасывайте PLL, чтобы выставить корректный фазовый сдвиг между входным и выходным тактовыми сигналами PLL.

На обоих входах `inclk0` и `inclk1` должны быть тактовые сигналы, прежде чем перевод сигнала `clkswitch` запустит процесс переключения. Не выполнение этого требования вызывает сбой в работе переключателя тактового сигнала.

- Приложения, в которых требуется средство переключения тактового сигнала и не значительный уход частоты, должны использовать низкочастотную PLL. Если меняется основной поступающий тактовый сигнал, низкочастотная PLL реагирует медленнее, чем высокочастотная. Когда происходит переключение, низкочастотная PLL передаёт сигнал остановки тактового сигнала на выходе медленнее, чем высокочастотная PLL. Низкочастотная PLL отфильтровывает джиттер основного тактового сигнала. Однако следует знать, что низкочастотная PLL также увеличивает время защёлки.
- После того, как произошло переключение, есть определённый фиксированный период ресинхронизации PLL, чтобы защёлкнуть новый тактовый сигнал. Это время зависит от конфигурации PLL.
- Если фазовый сдвиг между входным и выходным тактовыми сигналами PLL важен для вашего проекта, установите сигнал `areset` на 10 нс после выполнения переключения. Подождите появления сигнала защёлки (`locked`), прежде чем разрешать выходные тактовые сигналы PLL.

- На рис. 5-17 показано, как постепенно снижается частота VCO (ГУН), когда теряется основной тактовый сигнал, а затем, как увеличивается частота, когда VCO защёлкивается по второму тактовому сигналу. После защёлкивания VCO по второму тактовому сигналу, может происходить некоторая перегрузка (выход за пределы положенной частоты) VCO.

Figure 5-17. VCO Switchover Operating Frequency



- На время переключения запретите работу системе, если ваша система не толерантна к изменению частоты на время периода ресинхронизации PLL. Вы можете использовать статусные сигналы `clkbad[0]` и `clkbad[1]` для выключения PFD (`pfdena = 0`), чтобы VCO поддерживала последнюю рабочую частоту. Также вы можете использовать конечный автомат переключателя, чтобы переключиться на второй тактовый сигнал. После разрешения PFD, сигнал разрешения выходного тактового сигнала (`clkena`) может запретить тактовые выходы на время периода переключения и ресинхронизации. После стабилизации индикации защёлки, система может разрешить выходы тактовых сигналов.

## Программируемая полоса пропускания

Полоса пропускания PLL - это мерило способности PLL связывать входной тактовый сигнал с соответствующим ему джиттером. PLL семейства Cyclone III предлагают расширенные инструменты контроля над полосой пропускания PLL, используя программируемые характеристики обратной связи PLL, включения фильтра обратной связи и генератора подкачки. Коэффициент усиления обратной связи частоты в PLL на уровне 3 дБ определяет полосу пропускания PLL. Полоса пропускания примерно равна единице усиления для PLL с разомкнутой обратной связью.

## Реализация фазового сдвига

Фазовый сдвиг используется для реализации трудных задач по задержкам тактовых сигналов в чипах семейства Cyclone III. Фазовый сдвиг реализуется комбинацией фазы выхода VCO (ГУН) и временем запуска счётчика. Фаза выхода VCO и время запуска счётчика - является точным методом добавления задержек, поскольку они полностью вытекают из настроек счётчика, который, в свою очередь, не зависит от процессов, напряжения и температуры. Вы можете сдвигать фазу выходных тактовых сигналов на PLL семейства Cyclone III либо:

- с высоким разрешением, используя отвод фазы VCO, либо
- с низким разрешением, используя время запуска счётчика.



Высокое разрешение фазовых сдвигов реализуется разрешением для любых выходных счётчиков (C[4..0]) или для счётчика M использовать любую из восьми фаз VCO в качестве основного тактового сигнала. Это позволяет настраивать время задержки с высоким разрешением. В формуле 5-1 показано минимальное время задержки, которое вы можете получить, используя эту методику.

#### Equation 5-1. Fine Resolution Phase Shift

$$\Phi_{\text{fine}} = \frac{T_{\text{VCO}}}{8} = \frac{1}{8f_{\text{VCO}}} = \frac{N}{8Mf_{\text{REF}}}$$

##### Примечание к формуле 5-1:

(1)  $f_{\text{REF}}$  - это частота основного тактового сигнала на входе

Например, если  $f_{\text{REF}}$  равна 100 МГц,  $N = 1$ ,  $M = 8$ , то  $f_{\text{VCO}} = 800$  МГц, а  $\Phi_{\text{fine}} = 156.25$  ps. Рабочая частота PLL определяет этот фазовый сдвиг, а его значение зависит от частоты поступающего тактового сигнала и настроек счётчика.

Низкое разрешение фазового сдвига задаётся введением задержки перед запуском счётчика на предустановленное количество тактовых циклов счётчика. В формуле 5-2 показан фазовый сдвиг с низким разрешением.

#### Equation 5-2. Coarse Resolution Phase Shift

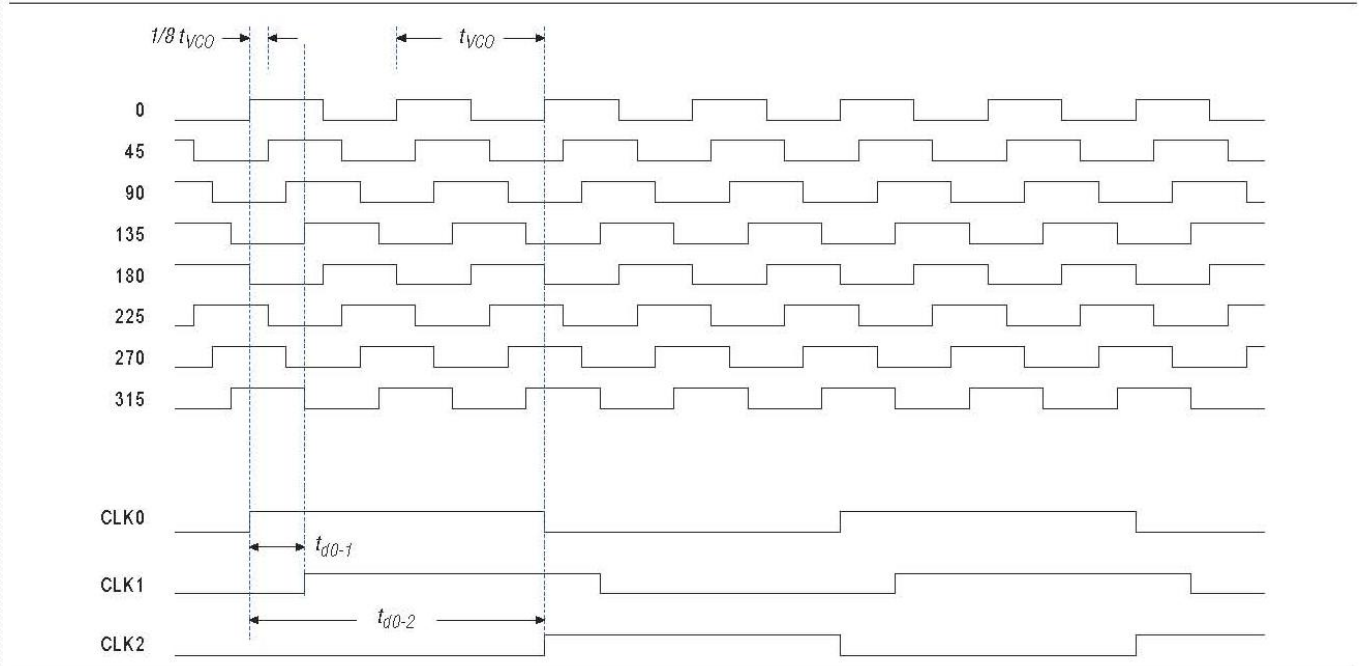
$$\Phi_{\text{coarse}} = \frac{C-1}{f_{\text{VCO}}} = \frac{(C-1)N}{Mf_{\text{REF}}}$$

##### Примечание к формуле 5-2:

(1)  $C$  - это заданное значение счёта для счётчика времени задержки - начальная настройка в разделе использования PLL в отчёте компилятора программы Quartus II. Начальное значение установлено в 1, что означает фазовый сдвиг  $C - 1 = 0^\circ$ .

На рис. 5-18 показан пример задания фазового сдвига высокого разрешения методом отвода фазы VCO. Восемь фаз VCO показаны и отмечены. В этом примере, CLK0 основан на  $0^\circ$  фазе VCO и имеет значение C, равное единице. Сигнал CLK0 делится на четыре, два периода VCO в единице и два периода в нуле. Сигнал CLK1 основан на  $135^\circ$  фазе VCO и имеет значение C, равное единице. Сигнал CLK1 также делится на четыре. Только в этом случае, два периода имеют начальное смещение на 3  $\Phi_{fine}$ . Сигнал CLK2 основан на  $0^\circ$  фазе VCO и имеет значение C, равное трём. Этим создаётся задержка на два  $\Phi_{coarse}$  (два полных периода VCO).

**Figure 5–18. Delay Insertion Using VCO Phase Output and Counter Delay Time**



Вы можете использовать низкое и высокое разрешение для фазового сдвига для реализации задержек тактового сигнала в чипах семейства Cyclone III.

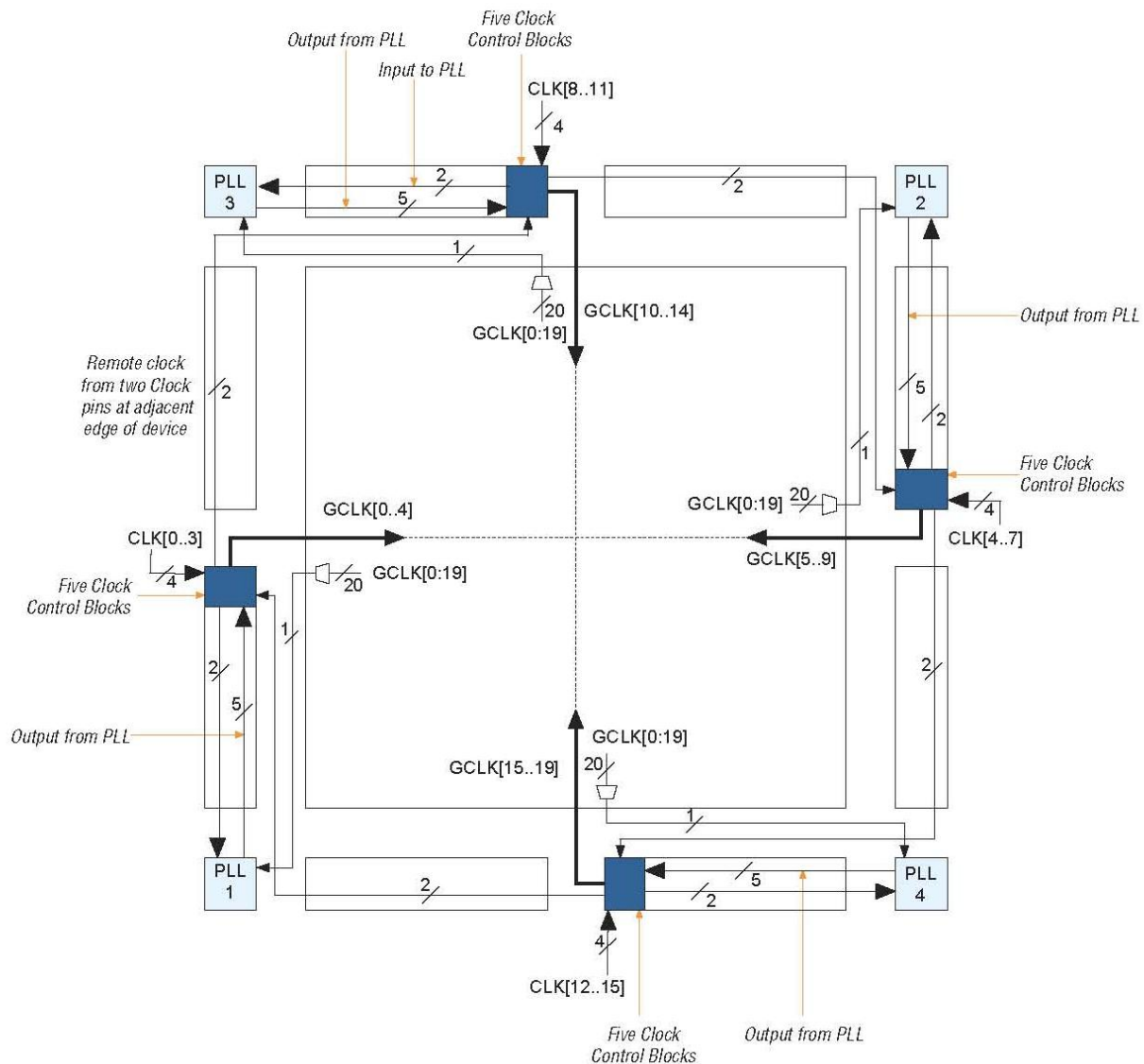
Семейство Cyclone III поддерживает динамический сдвиг фазы только отводов фазы VCO. Фазовый сдвиг может быть сконфигурирован в любое время. Каждый фазовый сдвиг занимает около одного периода  $\text{scanclock}$ , позволяя вам быстро реализовывать длительные фазовые сдвиги.

## Каскадирование PLL

Две PLL могут быть каскадно соединены друг с другом посредством тактовых сетей. Если в вашем проекте каскадируются PLL, то исходная (источник тактового сигнала) PLL должна иметь настройку low-bandwidth (низкочастотная), тогда как принимающая (приёмник тактового сигнала) PLL должна иметь настройку high-bandwidth (высокочастотная).

На рис. 5-19 показано использование GCLK для каскадирования PLL.

**Figure 5-19. PLL Cascading Using GCLK**



Следуйте рекомендациям при каскадировании PLL:

- Задайте первой PLL низкочастотную полосу пропускания, чтобы отфильтровать джиттер. Задайте второй PLL высокую полосу пропускания, чтобы отследить джиттер от первой PLL. Вы сможете увидеть в файле отчёта компилятора программы Quartus II, что полосы пропускания PLL не перекрываются. Если полосы пропускания перекрываются, в каскадной может возникнуть джиттер.

Вы можете вычислить расчётный джиттер PLL и статистическую погрешность фазы (SPE), используя временной анализатор TimeQuest в программе Quartus II. Используйте SDC команду "derive\_clock\_uncertainty", чтобы указать TimeQuest сгенерировать отчёт, называемый "PLLJ\_PLLSPE\_INFO.txt" в папке с вашим проектом. Затем используйте команды "set\_clock\_uncertainty", чтобы добавить значения джиттера и SPE к вашим ограничениям (constraints) для тактового сигнала.

- 
- Удерживайте вторую PLL в состоянии сброса, пока первая PLL не защёлкнется, чтобы выставить корректные настройки фазы для второй PLL.
  - Вы не сможете подключить любой из портов `inclk` любой PLL в каскадной схеме к выходу тактового сигнала PLL в каскадной схеме.