

Реконфигурация PLL

PLLs используют несколько делителей и различные отводы фазы VCO для выполнения синтеза частоты и фазовых сдвигов. В Cyclone III PLLs вы можете реконфигурировать и настройки счётчика и фазового сдвига выходного такта PLL в реальном времени. Также вы можете изменить компоненты генератора накачки и контурного фильтра, которые динамически влияют на полосу пропускания PLL.

Вы можете использовать эти компоненты PLL для обновления частоты выходного такта, полосы пропускания PLL и фазового сдвига в реальном времени — без полной реконфигурации FPGA.

Применение реконфигурации PLL в реальном времени особенно полезно в приложениях, которым приходится работать на нескольких частотах. Это также полезно при прототипировании окружающей среды, позволяя вам распределять частоты и динамически следить за фазой выходного такта. Например, система генерирует тестовые сигналы, требуемые для генерации и передачи сигналов на частотах 75 и 150 МГц, в зависимости от требований тестируемого прибора. Реконфигурация компонентов PLL в реальном времени позволяет вам переключаться между двумя такими выходными частотами за несколько микросекунд.

Вы можете также использовать это средство для слежения за задержками такта на выход (t_{CO}) в реальном времени, изменяя фазовый сдвиг выходного такта PLL. Это попытка избежать необходимости регенерировать конфигурационный файл с новыми настройками PLL.

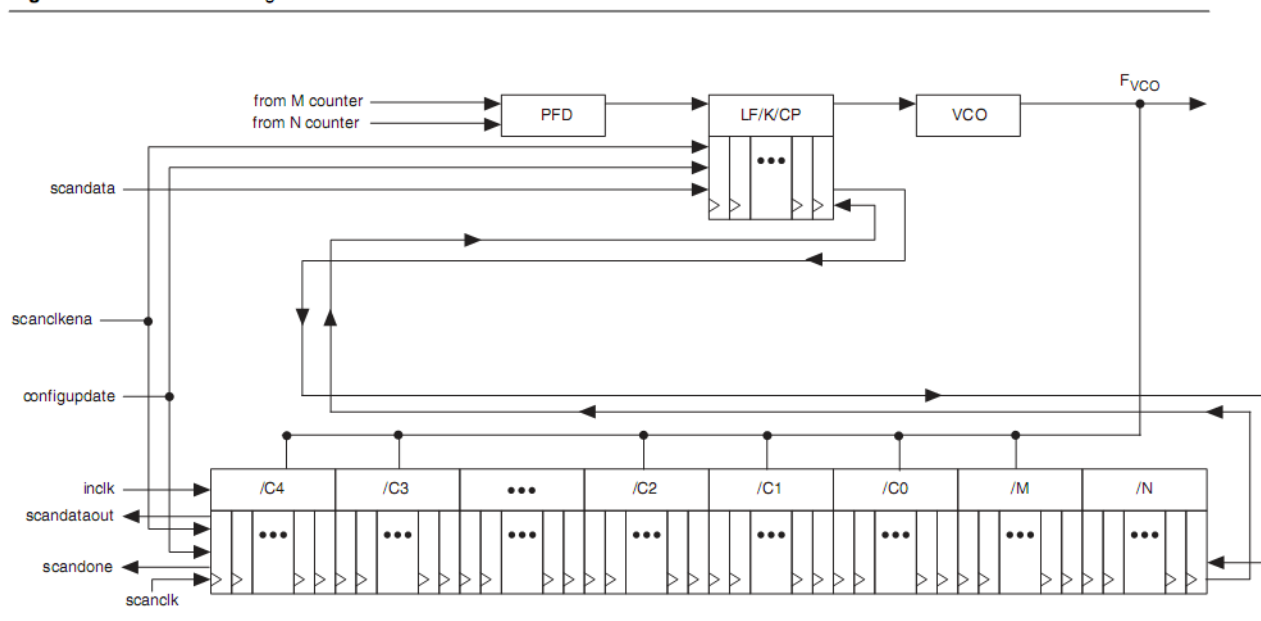
Аппаратная реализация реконфигурации PLL

Следующие компоненты PLL реконфигурируются в реальном времени:

- Пре-масштабирующий счётчик (N)
- Счётчик в петле обратной связи (M)
- Пост-масштабирующие счётчики (C0-C4)
- Динамическое регулирование тока генератора накачки (ICP) и компоненты петлевого фильтра (R, C) для содействия реконфигурации "на лету" полосы пропускания PLL.

На рисунке 6-25 показано, как регулируются динамически настройки счётчика PLL, сдвигая его новые настройки в цепи последовательного сдвигового регистра или цепи сканирования. Последовательный сдвиг данных в цепи сканирования через порт *scandata* и сдвиговые регистры тактируется *scanclk*. Максимальная частота *scanclk* — 100 МГц. После сдвига последнего бита данных, назначается сигнал *configupdate* длительностью не менее одного цикла такта *scanclk*, синхронно с обновлением битов конфигурации PLL данными из регистров сканирования.

Figure 6-25. PLL Reconfiguration Scan Chain



Настройки счётчика обновляются синхронно с частотой индивидуальных счётчиков. Поэтому не все счётчики обновляются одновременно.

В таблице 6-8 показано, как массив логики программируемой логической схемы (PLD) или выводы I/O управляют этими сигналами.

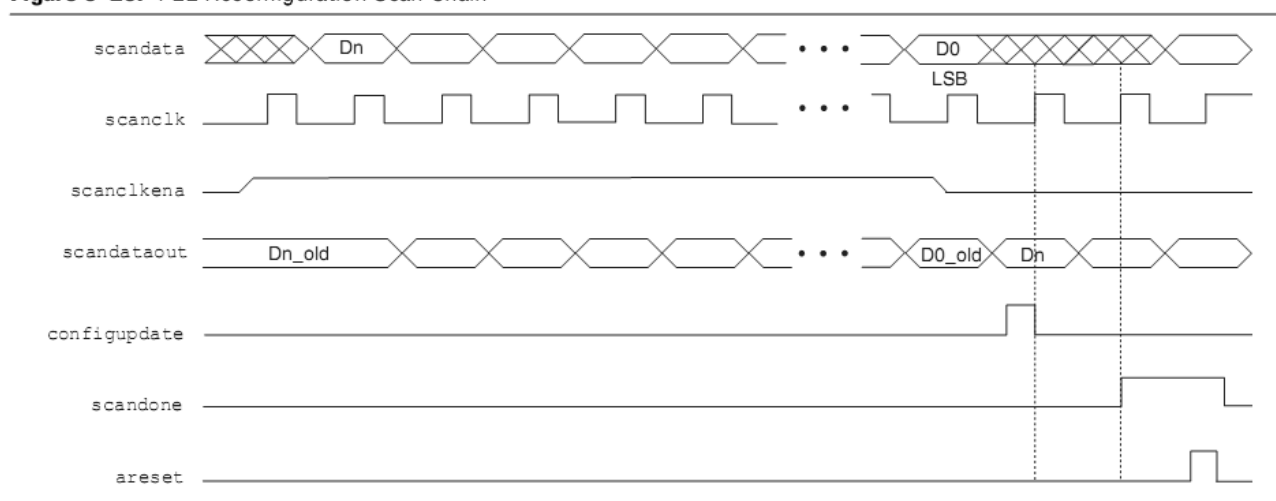
Таблица 6-8 Порты реконфигурации PLL в реальном времени

Имя порта PLL	Описание	Источник	Назначение
scandata	Последовательный вход потока данных в цепь сканирования	Массив логики или вывод I/O	Цепь реконфигурации PLL
scanclk	Сигнал последовательного такта. Этот такт может быть свободно запускаемым.	Массив логики или вывод I/O	Цепь реконфигурации PLL
configupdate	Обновляет данные из цепи сканирования в PLL. Активная 1.	Массив логики или вывод I/O	Цепь реконфигурации PLL
scanclkena	Разрешает <i>scanclk</i> и <i>scandata</i> для загрузки в цепь сканирования. Активный 1.	Массив логики или вывод I/O	Цепь реконфигурации PLL
scandone	Выдаёт единичный импульс, когда PLL завершила реконфигурацию.	Цепь реконфигурации PLL	Массив логики или вывод I/O
scandataout	Переносит содержимое из цепи сканирования.	Цепь реконфигурации PLL	Массив логики или вывод I/O

Для реконфигурации счётчиков PLL выполните следующие шаги:

1. Назначьте сигнал *scanclkena* на время не менее одного цикла *scanclk* прежде чем сдвигать первый бит *scandata* (Dn).
2. Последовательные данные (*scandata*) сдвигаются в цепь сканирования на втором нарастающем фронте *scanclk*.
3. После того, как все 144 бита были отсканированы в цепи сканирования, сигнал *scanclkena* снимается, чтобы избежать случайного сдвига бит в цепи сканирования.
4. Сигнал *configupdate* назначается на один период *scanclk* для обновления счётчиков PLL содержимым цепи сканирования.
5. Сигнал *scandone* переходит в "1", показывая, что PLL уже реконфигурирована. Фронт спада показывает, что счётчики PLL уже обновлены новыми настройками.
6. Сбросьте PLL, используя сигнал *areset*, если вы вносите некоторые изменения в настройки M, N счётчиков или в настройки ICP, R, C.
7. Вы можете повторять шаги 1 -5 для реконфигурации PLL любое количество раз.

На рисунке 6-26 показана функциональная симуляция средства реконфигурации PLL.

Figure 6–26. PLL Reconfiguration Scan Chain

Когда реконфигурируется тактовая частота счётчиков, соответствующие настройки счётчика фазового сдвига не могут реконфигурироваться, используя тот же самый интерфейс. Вы можете реконфигурировать фазовый сдвиг в реальном времени, используя интерфейс динамической реконфигурации фазового сдвига. Если вы реконфигурируете частоту счётчика, но хотите сохранить определённую (не нулевую) настройку фазового сдвига (например, 90°) по одному тактовому выходу, вам нужно будет реконфигурировать фазовый сдвиг после реконфигурации тактовой частоты счётчика.

Пост-масштабируемые счётчики (C0-C4)

Вы можете конфигурировать значения умножения или деления, а также рабочий цикл пост-масштабирующих счётчиков в реальном времени. Каждый счётчик имеет 8 бит настроек большего времени и 8 бит настроек меньшего времени. Рабочий цикл — это отношение большего или меньшего времени к общему времени цикла, который является суммой двух. Дополнительно, эти счётчики имеют два контрольных бита, *rbypass*, для шунтирования счётчика, и *rseledd*, для выбора рабочего цикла выходного такта.

Когда бит *rbypass* установлен в «1», он шунтирует счётчик, в результате — деление на 1. Когда этот бит установлен в «0», счётчики PLL эффективно делят выходную частоту VCO на счётчиках большего и меньшего времени. Например, если коэффициент пост-масштабирующего делителя равен 10, значения для большего и меньшего счётчиков установлены 5 и 5 соответственно, то достигается рабочий цикл 50-50%. PLL реализует этот рабочий цикл переходом выходного такта из единицы в нуль на нарастающем фронте выходного такта VCO. Однако, установки значений 4 и 6 для большего и меньшего счётчиков, могут создать выходной такт с рабочим циклом 40–60%.

Бит *rseledd* указывает на нечётный коэффициент деления выходной частоты VCO с 50% рабочим циклом. Например, если коэффициент пост-масштабирующего делителя равен 3, значения для большего и меньшего счётчиков установлены 2 и 1 соответственно, чтобы добиться этого деления. Это реализует рабочий цикл 67%–33%. Если вам нужен 50%–50% рабочий цикл, вы должны установить контрольный бит *rseledd* в «1», чтобы получать этот рабочий цикл независимо от нечётного коэффициента. PLL реализует этот рабочий цикл переходом выходного такта из единицы в нуль на спадающем фронте выходного такта VCO. Когда вы устанавливаете *rseledd* = 1, выделяется 0,5 периода из большего времени и добавляется 0,5 периода меньшему времени. Например:

- Счётчик большего времени = 2 периода
- Счётчик меньшего времени = 1 период

■ $rseledd = 1$ это эквивалентно:

- Счётчик большего времени = 1,5 периода
- Счётчик меньшего времени = 1,5 периода
- Рабочий цикл = $(1.5/3) \%$ счётчика большего времени и $(1.5/3) \%$ счётчика меньшего времени

Описание цепи сканирования

Cyclone III PLLs имеют 144-битную цепь сканирования. В таблице 6-9 показано количество бит для каждого компонента PLL.

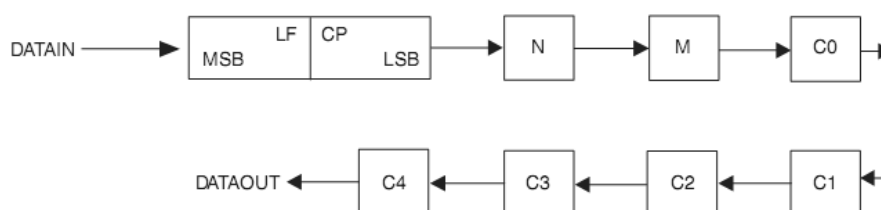
Таблица 6-9. Биты перепрограммирования PLL

Название блока	Численность битов		
	Счетчик	Другие	Всего
C4 (1)	16	2 (2)	18
C3	16	2 (2)	18
C2	16	2 (2)	18
C1	16	2 (2)	18
C0	16	2 (2)	18
M	16	2 (2)	18
N	16	2 (2)	18
Генератор накачки	9	0	9
Петлевой фильтр (3)	9	0	9
ВСЕГО			144

Примечания к таблице 6-9:

- (1) LSB бит для значения низкого счёта C4 — это первый бит, сдвигаемый в цепь сканирования.
- (2) Это два контрольных бита *rbypass* для шунтирования счётчика и *rseledd* для выбора выходного такта.
- (3) MSB для петлевого фильтра — это последний бит, сдвигаемый в цепь сканирования.

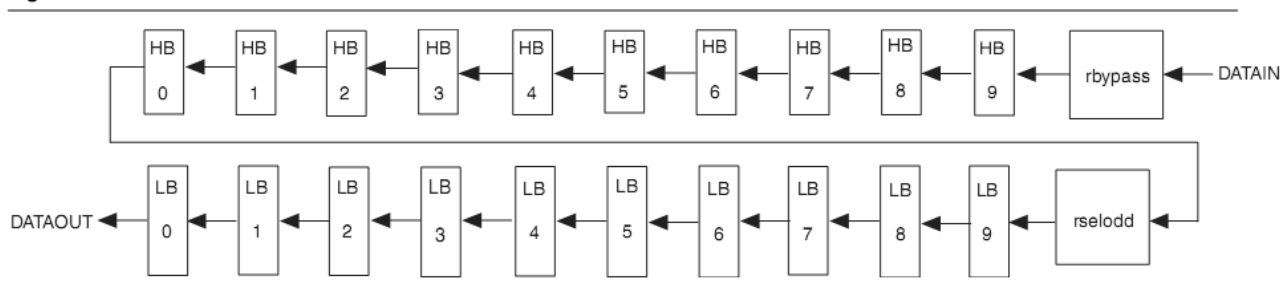
Figure 6-27. PLL Component Scan Chain Order



На рисунке 6-27 показан порядок следования компонентов в цепи сканирования.

На рисунке 6-28 показан порядок следования битов для одного пост-масштабируемого счётчика PLL в Cyclone III PLLs.

Figure 6–28. Scan Chain Bit Order



Генератор накачки и петлевого фильтра

Вы можете реконфигурировать настройки генератора накачки и петлевого фильтра для обновления полосы пропускания PLL в реальном времени. В таблицах 6-10 — 6-12 показаны возможные настройки для генератора накачки (Icp) и петлевого фильтра, значения резистора (R) и конденсатора (C), в Cyclone III PLLs.

Таблица 6-10 Изменение контрольных битов накачки

CP[2]	CP[1]	CP[0]	Установка (дес.)
0	0	0	0
1	0	0	1
1	1	0	3
1	1	1	7

Таблица 6-11 Контроль сопротивления резистора в петлевом фильтре

LFR[4]	LFR[3]	LFR[2]	LFR[1]	LFR[0]	Установка (дес.)
0	0	0	0	0	0
0	0	0	1	1	3
0	0	1	0	0	4
0	1	0	0	0	8
1	0	0	0	0	16
1	0	0	1	1	19
1	0	1	0	0	20
1	1	0	0	0	24
1	1	0	1	1	27
1	1	1	0	0	28
1	1	1	1	0	30

Таблица 6-12 Контроль высокочастотного конденсатора в петлевом фильтре

LFC[1]	LFC[0]	Установка (дес.)
0	0	0
0	1	1
1	1	3

Шунтирование PLL счётчиков

Шунтирование PLL счётчиков даёт в результате коэффициенты единицы для умножающего (счётчика М) и делящего (N, C0 - C4 счётчики) счётчиков.

В таблице 6-13 показаны настройки для шунтирования счётчиков Cyclone III PLLs.

Таблица 6-13 Настройки счетчика

Биты установки цепи сканирования PLL									Описание
LSB								MSB	
X	X	X	X	X	X	X	X	1 (1)	Счетчики PLL шунтированы
X	X	X	X	X	X	X	X	0 (1)	Счетчики PLL не шунтированы

Примечания к таблице 6-13:

1. Бит шунтирования.

Для шунтирования любого счетчика PLL необходимо установить бит пропуска в 1. Этим игнорируются значения остальных битов.

Динамический сдвиг фазы

Средство динамического сдвига фазы позволяет вам динамично регулировать выходную фазу отдельных выходов PLL по отношению к другим выходам и поступающему в PLL такту, без необходимости посылки последовательных данных через цепь сканирования соответствующей PLL. Это упрощает интерфейс и позволяет вам быстро регулировать задержки от такта на выход (tco), изменяя фазовый сдвиг выходного такта в реальном времени. Это достигается приращением или уменьшением секций фазовых отводов VCO, поступающих на С счётчик или на М счётчик. Фаза сдвигается на 1/8 частоты VCO за раз. Выходные такты остаются активными во время процесса реконфигурации.

В таблице 6-14 показаны контрольные сигналы, которые используются для динамического сдвига фазы.

Таблица 6-14 Контрольные сигналы динамического изменения фазы (часть 1 из 2)

Название сигнала	Описание	Источник	Назначение
PHASECOUNT RSELECT[2..0]	Выбор счетчика. Три бита декодируются для выбора только М или одного из С счетчиков для регулирования фазы. Одна карта адресов для выбора всех С счетчиков. Этот сигнал регистрируется в PLL по нарастающему фронту SCANCLK.	Логический массив или I/O выводы	Схема реконфигурации PLL
PHASEUPDOWN	Выбор направления динамического фазового сдвига: 1 – увеличение, 0 – уменьшение. Этот сигнал регистрируется в PLL по нарастающему фронту SCANCLK.	Логический массив или I/O выводы	Схема реконфигурации PLL
PHASESTEP	Логическая 1 разрешает динамический фазовый сдвиг.	Логический массив или I/O выводы	Схема реконфигурации PLL

Таблица 6-14 Контрольные сигналы динамического изменения фазы (часть 2 из 2)

Название сигнала	Описание	Источник	Назначение
SCANCLK	Свободно запущенные такты ядра, используются в комбинации с PHASESTEP для разрешения или запрещения динамического сдвига фазы. Используйте для динамической реконфигурации.	GCLK или I/O выводы	Схема реконфигурации PLL
PHASEDONE	Когда установлено, показывает логике ядра, что регулировка фазы закончена и PLL готова к действию на второй возможный такт регулирования. Установка основана на внутреннем таймере PLL. Дезактивация по фронту спада SCANCLK	Схема реконфигурации PLL	Логический массив или I/O выводы

В таблице 6-15 показан выбор PLL счётчика, основанный на соответствующих настройках PHASECOUNTERSELECT.

Таблица 6-15 Карта выбора фазовых счетчиков

PHASECOUNTERSELECT[2]	[1]	[0]	Выбор
0	0	0	Все выходные счетчики
0	0	1	M
0	1	0	C0
0	1	1	C1
1	0	0	C2
1	0	1	C3
1	1	0	C4

Для выполнения одного шага фазового сдвига, выполните следующую процедуру:

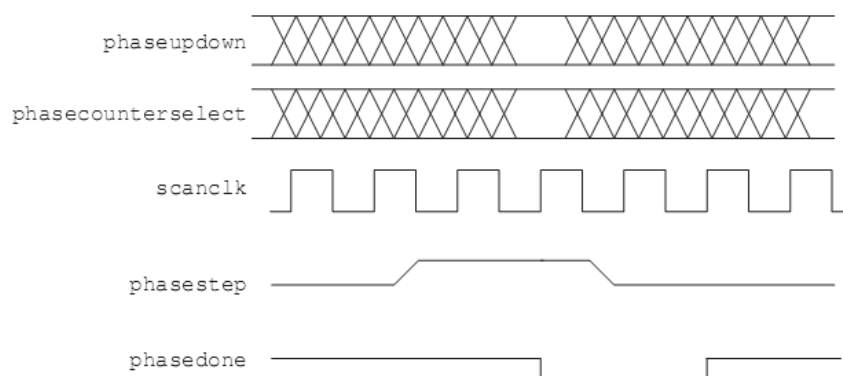
1. Установите, как вам нужно, сигналы *phaseupdown* и *phasecounterselect*
2. Назначьте *phasetestep* не менее двух периодов *scanclk*. Каждый импульс *phasetestep* разрешает один фазовый сдвиг.
3. Снимите *phasetestep*.
4. Подождите, пока *phasedone* установится в "1".
5. Вы можете повторять пункты с 1 по 4 столько раз, сколько потребуется, чтобы получить различные сдвиги фазы.

Все сигналы синхронны *scanclk*, потому что защёлкиваются по фронтам *scanclk* и должны достигать требований *tsu/th* (по отношению к фронтам *scanclk*).

Динамический сдвиг фазы может повторяться бесконечно.

Сигнал *phasetest* защёлкивается на отрицательном фронте *scanclk*. На рисунке 6-29 это показано на втором фронте спада *scanclk*. *Phasetest* должен оставаться в "1" не менее двух периодов *scanclk*. На втором нарастающем фронте *scanclk* после защёлкивания *phasetest* (показано на четвёртом нарастающем фронте) значения *phaseupdown* и *phasecounterselect* защёлкиваются и PLL начинает динамический сдвиг фазы для определённого счётчика или счётчиков и в выбранном направлении. На четвёртом нарастающем фронте *scanclk*, сигнал *phasedone* переходит из "1" в "0" и сохраняет "0", пока PLL не закончит фазовый сдвиг. Вы можете выполнить другой динамический фазовый сдвиг после того, как сигнал *phasedone* перейдёт из "0" в "1".

Figure 6–29. PLL Dynamic Phase Shift



В зависимости от частот VCO и *scanclk*, время, когда *phasedone* равен "0", может быть больше или равно одному периоду *scanclk*. Максимальное время динамического реконфигурирования фазы может быть определено (TBD), основываясь на характеристиках чипа.

После того, как *phasedone* переходит из "0" в "1", вы можете выполнить другой динамический сдвиг фазы. Импульсы *phasetest* должны быть по отдельности не менее одного периода *scanclk*.

За подробной информацией о менеджере плагинов MegaWizard ALTPLL_RECONFIG, обратитесь к руководству пользователя мегафункцией ALTPLL_RECONFIG.