

УДК 51:33(075.8)

## Оценка сложности самосинхронных логических элементов FPGA

А. Ю. Скорнякова

Пермский национальный исследовательский политехнический университет  
Россия, 614990, г. Пермь, ул. Комсомольский пр., 29  
juris-plot@mail.ru; +7-951-949-65-22

В настоящее время широко используются так называемые программируемые логические интегральные схемы (ПЛИС). Они, как правило, используют синхронную обработку информации, основанную на тактовом генераторе, причем тактовая частота рассчитывается на самый худший случай, на самую большую длительность переходного процесса. Асинхронные ПЛИС работают по реальным задержкам элементов и устройств, однако синтез асинхронных схем значительно сложнее, чем синхронных. Одним из вариантов асинхронных схем являются так называемые самосинхронные (СС) схемы (ССС). В статье рассматриваются предложенные логические элементы для СС ПЛИС типа FPGA и приводятся полученные оценки сложности.

**Ключевые слова:** самосинхронные схемы; FPGA; LUT; сложность.

DOI: 10.17072/1993-0550-2019-4-86-89

### Введение

Одной из важнейших задач синтеза цифровых автоматов является задача учета переходных процессов в элементах и устройствах, которые так же, как и негативные внешние факторы, являются причиной сбоев и отказов систем управления и вычислительной техники.

В настоящее время наиболее широко используется синхронная реализация цифровых автоматов. Период следования импульсов тактового генератора выбирают таким, чтобы комбинационная схема "успевала" переключиться до прихода очередного импульса. Для последовательностных устройств (автоматов) используют регистр по входу и выходу и двухфазную синхронизацию.

В первой фазе во входной регистр записывают входной набор (вектор), после чего начинается обработка в логическом преобразователе автомата (вычисляются функции переходов и выходов). В это время значение входного регистра не изменяется, новая информация не принимается. После того, как все переходные процессы в логическом преобразователе гарантированно закончатся (учитывается самый худший, то есть самый продолжительный процесс), по импульсу второй фазы выходная информация (вектор) записывается в выходной регистр. Асинхронные схемы не используют "общие часы" тактового генератора, но должны учитывать все нюансы переходных процессов, все состязания (гонки) входов, цепей, элементов памяти при всех возможных комбинациях входных сигналов. Понятно, что в полной мере это сделать невозможно, поэтому синтез таких схем неизмеримо более сложный, чем синхронных.

Самосинхронные схемы ССС [1–3] используют устройства фиксации завершения переходных процессов (индикаторы, гистерезисные триггеры Г-триггеры). ССС работают по реальным задержкам, что улучшает быстродействие, но они, в свою очередь, снижают этот эффект за счет введения своей двухфазной дисциплины. Имеется фаза гашения (спейсер) и рабочая фаза [1–3]. Аппаратные затраты для комбинационной логики увеличиваются более чем вдвое. Введение каскадирования индикаторов и Г-триггеров для схем большой размерности подчас приводит к тому, что быстродействие ССС становится даже хуже, чем синхронных.

Самосинхронные схемы ССС [1–3] используют устройства фиксации завершения переходных процессов (индикаторы, гистерезисные триггеры Г-триггеры). ССС работают по реальным задержкам, что улучшает быстродействие, но они, в свою очередь, снижают этот эффект за счет введения своей двухфазной дисциплины. Имеется фаза гашения (спейсер) и рабочая фаза [1–3]. Аппаратные затраты для комбинационной логики увеличиваются более чем вдвое. Введение каскадирования индикаторов и Г-триггеров для схем большой размерности подчас приводит к тому, что быстродействие ССС становится даже хуже, чем синхронных.

Однако ССС могут быть востребованы при переходе к нанoeлектронике, когда уже начинают сказываться квантовые эффекты, они обеспечивают повышение надежности обработки информации и перспективны в области энергоэффективности. Особенно актуально сочетание СС-принципа и гибкости ПЛИС.

### 1. Элемент ST 1-LUT

Предлагается модификация известного элемента 1-LUT [5, 6], изображенная на рис. 1.

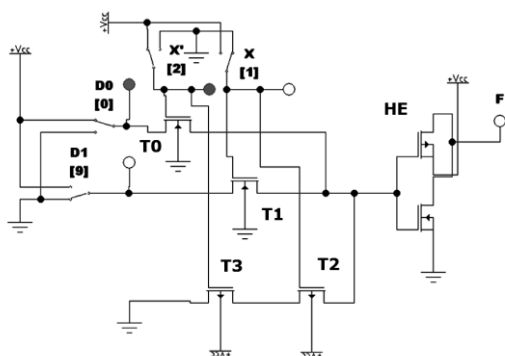


Рис. 1. Элемент ST 1-LUT, один канал

В известном 1-LUT имеются передающие транзисторы T0 и T1. В зависимости от значения переменной X (либо 0, либо 1), которая управляет затворами транзисторов на вход инвертора-элемента НЕ поступает либо информация d0, либо d1. Это так называемая конфигурационная информация (настройка), которая в ПЛИС записывается в ячейки статической оперативной памяти (SRAM). В них записывается таблица истинности заданной логической функции, в данном случае одной переменной. Поэтому на выходе F активируется требуемая функция. Строго говоря, таблица истинности в нашем случае должна быть инверсной (противоположной) заданной.

В настоящем логическом элементе на входах передающих транзисторов стоят также инверторы (элементы НЕ, они выполняют роль своего рода усилителей). Инверторы также установлены по входам переменных так, что всегда имеется переменная и ее отрицание.

На рис. 1 такого инвертора нет, а использован отдельный вход X', поскольку в ССС переменные так и подаются – это называется *парафазный код*. В рабочей фазе X и X' инверсны (противоположны). В фазе гашения они равны, это и позволяет зафиксировать факт окончания переходного процесса.

В предлагаемом логическом элементе для реализации гашения вводятся транзисторы T2 и T3. В случае  $X = X' = 1$  на вход НЕ поступает 0 с шины "Ноль вольт", на выходе F устанавливается 1, что и фиксирует завершение фазы гашения (спейсера). Однако для реализации ССС устройства, приведенного на рис. 1 недостаточно. Необходимо еще такое же почти устройство (так называемый *двойственный канал*), реализующий двойственную логическую функцию, – это необходимо для получения на выходах F и F' (выход двойственного канала) инверсных значений, наличие которых фиксирует завершение рабочей фазы на индикаторе, который не указан на рис. 1 (по существу это обычный элемент 2И-НЕ. Значение 1 на его выходе фиксирует Г-триггер, также не указанный на рис. 1. Потом происходит гашение, оба выхода F и F' устанавливаются в 1 и на выходе индикатора 2И-НЕ формируется 0, что принимается Г-триггером, как завершение фазы гашения. Г-триггер "собирает" сигналы нескольких индикаторов с нескольких устройств. Выход его устанавливается в 1, когда все входы стали 1 (и его в том числе).

Таким образом, подается команда на подачу очередного входного набора.

Далее ожидается переход всех входов в 0 (завершение рабочей фазы) и так далее. Моделирование элемента 1-LUT-ST в рабочей фазе и фазе гашения представлено на рис. 2–4.

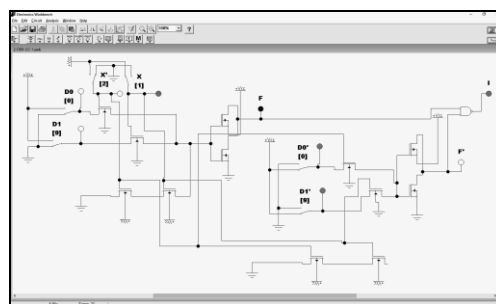


Рис. 2. Элемент ST 1-LUT, два канала в рабочей фазе, D0=0, D1=0

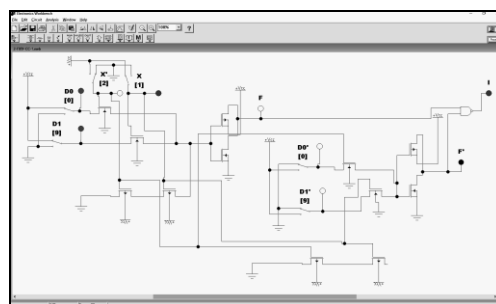


Рис. 3. Элемент ST 1-LUT, два канала в рабочей фазе, D0=1, D1=1

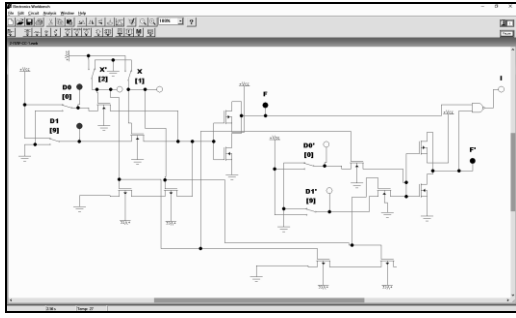


Рис. 4. Элемент ST 1-LUT, два канала в фазе спейсера

## 2. Оценки сложности ST LUT

Получим оценки сложности ST LUT для  $n$  переменных – выражение (1).

$$L_{n,k,ST} = 2 \cdot \{2^{[n]} \cdot 8 + (2^{[k]+1} + 6[k]) \cdot \sum_{i=1}^{[n]} 2^{[n]-[i][k]} + (2^{[n]-[k]} \cdot [k]^{[k]+1} + 6 \cdot \left( [n] - \left\lfloor \frac{[n]}{[k]} \right\rfloor \cdot [k] \right) + 6[n] + 2 \cdot [n] \} + 2 \cdot \sum_{i=1}^{[n]} 2^{[n]-[i][k]} + 2 \cdot \left[ \left\lfloor \frac{[n]}{[k]} \right\rfloor - \left\lfloor \frac{[n]}{[k]} \right\rfloor \right]. \quad (1)$$

В (1)  $2^{[k]+1} + 6[k]$  – сложность одного элементарного  $k$ -дерева  $[...]$ -округление в нижнюю сторону (floor); таких деревьев ( $k$ -LUT) необходимо в первом слое  $2^{[n]-[k]}$ , затем нужно провести декомпозицию  $k$ -LUT этого первого слоя, получаем  $2^{[n]-[k]-[k]}$

Всего необходимо  $i$   $k$ -LUT, где  $i$  определяется из соотношения  $[i] = \left\lfloor \frac{[n]}{[k]} \right\rfloor$  всего

$$\sum_{i=1}^{\left\lfloor \frac{[n]}{[k]} \right\rfloor} 2^{[n]-[i][k]} \text{ и последний LUT на } [n] - \left\lfloor \frac{[n]}{[k]} \right\rfloor \cdot [k]$$

переменных.

$2 \cdot [n]$ -сложность дополнительных цепочек

спейсера,  $2 \cdot \sum_{i=1}^{\left\lfloor \frac{[n]}{[k]} \right\rfloor} 2^{[n]-[i][k]}$ -сложность индикаторов  $k$ -LUT + индикатор последнего LUT

$$2 \cdot \left[ \left\lfloor \frac{[n]}{[k]} \right\rfloor - \left\lfloor \frac{[n]}{[k]} \right\rfloor \right].$$

Выражение (1) не учитывает затраты на фиксацию переходного процесса Г-триггерами. Учесть двухвходовые Г-триггеры сложностью 12 транзисторов можно, положив  $k=2$ , а  $n$ :

$$n = \sum_{i=1}^{\left\lfloor \frac{[n]}{[k]} \right\rfloor} 2^{[n]-[i][k]} + \left[ \frac{[n]}{[k]} \right] - \left\lfloor \frac{[n]}{[k]} \right\rfloor. \quad (2)$$

Таким образом, получаем сложность Г-триггеров:

$$12 \cdot \left\{ \sum_{i=1}^{\left\lfloor \frac{[n]}{[k]} \right\rfloor} 2^{[n]-[i][k]} \cdot \left[ \frac{[n]}{[k]} \right] - \left\lfloor \frac{[n]}{[k]} \right\rfloor \right\} + \left[ \frac{[n]}{[k]} \right] - \left\lfloor \frac{[n]}{[k]} \right\rfloor. \quad (3)$$

Временная задержка LUT-ST в количестве транзисторов (без учета задержки Г-триггеров) увеличивается на задержку индикаторов:

$$T_{n,k,ST} = n + 2 \left[ \frac{[n]}{[k]} \right] + 2 \cdot \sum_{i=1}^{\left\lfloor \frac{[n]}{[k]} \right\rfloor} 2^{[n]-[i][k]} + 2 \cdot \left[ \left\lfloor \frac{[n]}{[k]} \right\rfloor - \left\lfloor \frac{[n]}{[k]} \right\rfloor \right]. \quad (4)$$

Графики сравнения сложности LUT и ST LUT на  $n$  переменных ( $n=3...7$ ) представлены на рис. 5.

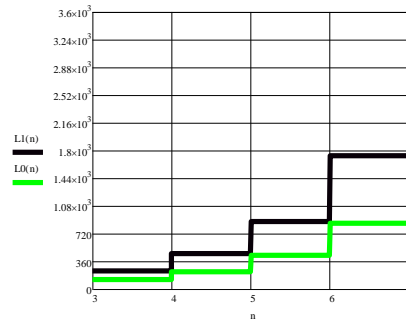


Рис. 5. Графики сравнения сложности LUT и ST LUT на  $n$  переменных

## Выводы

Таким образом, получены оценки сложности предлагаемого ST LUT с учетом декомпозиции по  $k$  (элементарному дереву на  $k$  переменных). Построены графики сравнения с исходной сложностью LUT. Имеется более чем двукратное увеличение сложности по сравнению с синхронным вариантом. Однако такое увеличение может быть оправданным для увеличения надежности.

В дальнейшем целесообразно провести топологическое моделирование и уточнить оценки по площади кристалла, энергопотреблению и максимальной задержке.

### Список литературы

1. *Апериодические* автоматы / под ред. В.И. Варшавского. М.: Наука, 1976. С. 304.
2. *Основы самосинхронных электронных схем* [Электрон. ресурс] / Л.П. Плеханов. Эл. изд. М.: БИНОМ. Лаборатория знаний, 2013. 208 с.
3. *Степченков Ю.А., Денисов А.Н., Дьяченко Ю.Г., Гринфельд Ф.И., Филимонок О.П., Морозов Н.В., Степченков Д.Ю.* Библиотека элементов для проектирования самосинхронных полузаказных микросхем серий 5503/5507 и 5508/5509. М.: ИПИ РАН, 2012. 1348 с.
4. *Степченков Ю.А. и др.* Самосинхронный вычислитель для высоконадежных применений // Всерос. науч.-техн. конф. "Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС)": сб. тр. Институт проблем проектирования в микроэлектронике РАН, 2010. № 1.
5. *Тюрин С.Ф.* Энергоэффективные LUT FPGA, концепция, модели, оценки. Наноиндустрия. 2017. № 4 (74). С. 194–200.
6. *Тюрин С.Ф.* Скользящее резервирование толерантных элементов. Надежность. 2017. Т. 17, № 1 (60). С. 17–21.
7. *Тюрин С.Ф., Скорнякова А.Ю.* Программируемое логическое устройство. Патент на изобретение RUS 2653304. Опубл. 07.05.2018, Бюл. № 13.
8. *Тюрин С.Ф., Скорнякова А.Ю.* Программируемое логическое устройство. Патент на изобретение RUS 2653301. Опубл. 07.05.2018, Бюл. № 13.
9. *Тюрин С.Ф., Скорнякова А.Ю.* Универсальный логический элемент для самосинхронной схемы // Вестник Рязанского государственного радиотехнического университета. 2017. № 61. С. 41–45.
10. *Тюрин С.Ф., Скорнякова А.Ю.* Самосинхронный универсальный логический элемент для реализации систем функций // Инженерный вестник Дона. 2017. № 1.
11. *Skornyakova A.Yu.* Statement of the problem of finding an optimal set of complete tolerant Boolean functions in the synthesis of self-timed circuits. В сборнике: Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, EIConRus. 2018. С. 244–246.
12. *Тюрин С.Ф., Никитин А.С., Вихорев Р.В., Скорнякова А.Ю.* Выбор набора конфигурируемых логических элементов с использованием венгерского метода // Вестник Пермского университета. Математика. Механика. Информатика. 2017. Вып. 2(37). С. 65–68.

## Evaluation of FPGA's self-timed LUT complexity

**A. Yu. Skornyakova**

Perm National Research Polytechnic University; 29, Komsomolsky prospekt, Perm, 614990, Russia  
juris-plot@mail.ru; +7-951-949-65-22

Currently, the so-called programmable logic integrated circuits (including FPGAs) are widely applied. They usually use synchronized information processing based on a clock generator, with the clock frequency calculated for the worst case possible, for the longest transition process. Asynchronous FPGAs work according to real delays of elements and devices; however, the synthesis of asynchronous circuits is much more complicated than that of synchronous ones. One of the types of asynchronous circuits are the so-called self-timed (ST) circuits (STC). The article discusses the proposed logical elements for ST FPGA and provides the obtained estimates of complexity.

**Keywords:** *self-timed circuits; FPGA; LUT; complexity.*