УДК 621.385.2

Анализ избыточности битовой последовательности для проектов программируемых логических интегральных схем

Панкратов А. В.*, Якимов В. Л., Маковский В. Н***.**

Военно-космическая академия имени А. Ф. Можайского, Ждановская набережная, 13, Санкт-Петербург, 197082, Россия

**e-mail: pankratov-av@rambler.ru*

***e-mail: yakim78@yandex.ru*

****e-mail: sla_mvn777@mail.ru*

Аннотация

Рассмотрен механизм отображения высокоуровневого описания цифрового устройства на конфигурирующую битовую последовательность при проектировании ПЛИС. Показана возможность выделения семейства битовых последовательностей, соответствующих заданной схеме. Приведен пример неустранимой избыточности конфигурирующей битовой последовательности для узлов коммутирующих матриц.

Ключевые слова: проектирование программируемых логических интегральных схем, битовый конфигурационный поток, коммутирующие матрицы.

Введение

В настоящее время сильно расширился спектр применения программируемых логических интегральных схем (ПЛИС), в основном из-за роста их емкости и быстродействия. На основе ПЛИС можно создавать практически любые цифровые схемы, любого уровня сложности с интеграцией промежуточной и сопрягающей логики. Программируемые логические интегральные схемы состоят из логических блоков, коммутирующих путей, элементов ввода-вывода. Особенностью современных ПЛИС является наличие схем управления конфигурацией через встроенные последовательные протоколы. Конфигурирующая битовая последовательность полностью определяет функционал ПЛИС, поэтому представляет интерес для исследования.

Структура битовой последовательности

Конфигурирующая битовая последовательность определяет состояние триггеров, управляющих внутренними связями и регистров, задающих функционал логических блоков. Внутри логических блоков логические элементы соединяются посредством локальной программируемой матрицы соединений, позволяющей соединять любые логические элементы. Логические блоки связаны между собой и с элементами ввода – вывода посредством глобальной программируемой матрицы соединений. Стоит отметить, что основная часть конфигурирующей битовой последовательности отвечает именно за конфигурацию матриц соединений.

На рисунке 1 показан упрощенный механизм синтеза конфигурирующей битовой последовательности. На каждом этапе синтеза используется дополнительная информация об аппаратной части, происходит переход на более низкий уровень описания и привязка к ресурсам микросхемы.

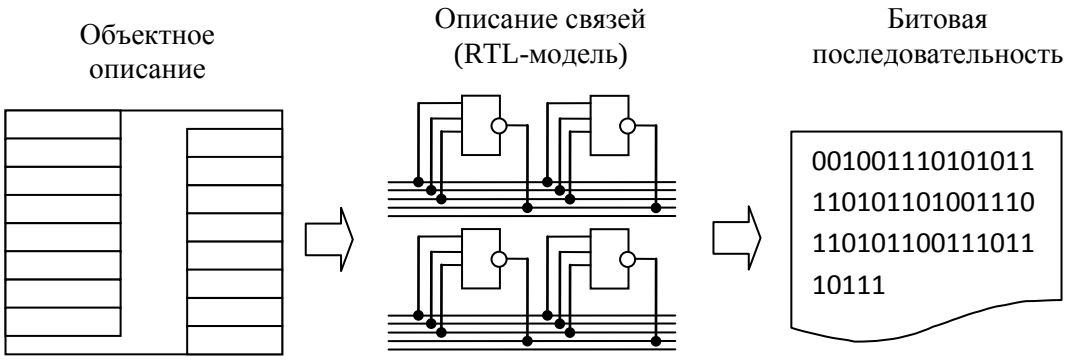

Рис. 1. Механизм синтеза конфигурирующей последовательности.

Для описанного механизма синтеза предполагается множественность отображений при переходе на следующий уровень, а выбор конкретного отображения осуществляется системой проектирования с учетом пожеланий и ограничений. Множественность отображений RTL-модели на битовую последовательность возможна только в случае некоторой избыточности последней.

Оценка избыточности битовой последовательности

Для оценки избыточности конфигурирующей битовой последовательности рассмотрим сопряжение её участка с узлом коммутации программируемой матрицы соединений. На рисунке 2 показан типовой узел

коммутации программируемой матрицы соединений, сопряженный с участком конфигурирующей битовой последовательности. Здесь стоит заметить, что следование конфигурирующих битов привязано к топологии микросхемы, а не к логическим элементам, а показанную на рисунке 2 последовательность следует рассматривать как перегруппировку по логическим элементам. Коммутация узла происходит посредством шести управляемых битами конфигурации вентилей M1, M2, … , M6. Число возможных комбинаций управляющих битов для одного узла составляет 64, что превышает возможности коммутации узла. Так, для указанного узла возможно 15 состояний коммутации, соответствующих всем комбинациям соединения четырех проводников, включая разрыв всех цепей и соединение в один общий узел.

Для оценки нижней границы избыточности битовой последовательности можно воспользоваться следующим выражением:

$$inf(L_k) = 1 - \frac{\log_2 N_k}{m}, \qquad (1)$$

где относительная избыточность L_k соответствует доле неизбежно избыточной информации в m-битном потоке конфигурации, задающем N_k возможных внутренних состояний.

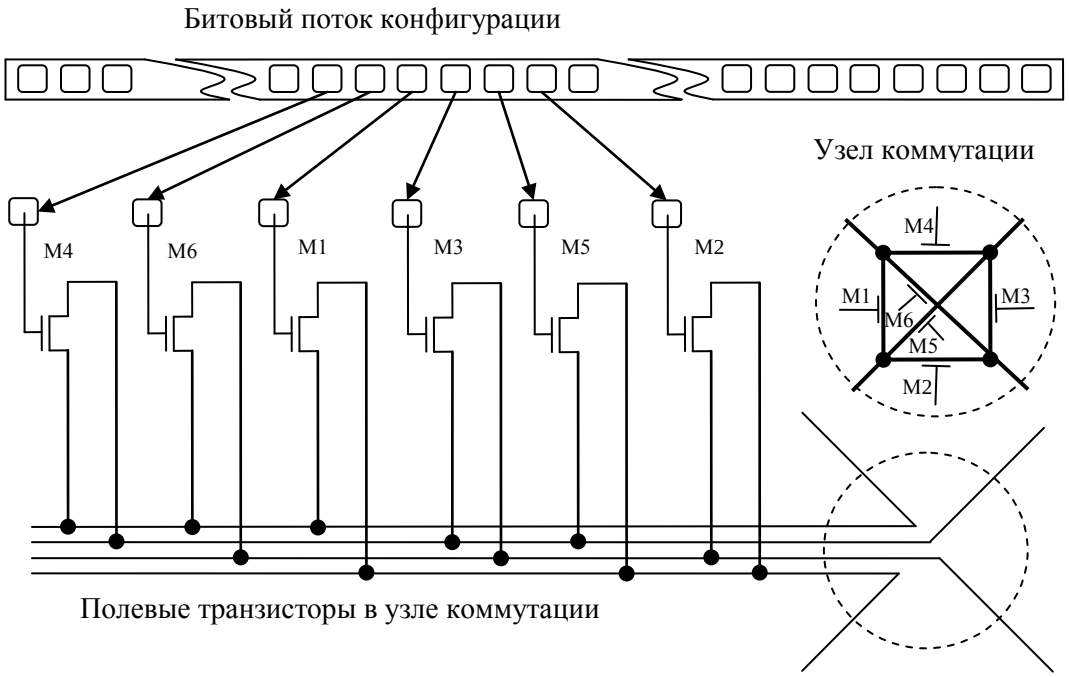

Рис. 2. Сопряжение битового потока с узлом коммутации

Следует отметить, что избыточность характерна не только для коммутирующих матриц. Логические блоки ПЛИС, задействованные, например, для реализации микропроцессорных модулей, используют комбинационные таблицы не более чем на треть. Ввиду малой доли битовой последовательности, конфигурирующей логические блоки, результирующая избыточность практически не зависит от интенсивности их использования. Так, для микросхемы xc3s1200e битовая последовательность составляет 3841904 бита при количестве логических блоков 2168 [1, 2]. Часть битовой последовательности, конфигурирующая логические блоки не превышает 5% для указанной микросхемы. Нижняя граница избыточности, рассчитанная по выражению (1) составит $inf(L_k)$ ≈35%.

Несмотря на успешное развитие рынка микроэлектроники, которое многим обязано ПЛИС, как наиболее универсальным инструментам отладки, задействовать все ресурсы современных микросхем не представляется возможным в силу описанной избыточности внутренних связей. Можно проследить аналогию между разработкой современного программного обеспечения и разработкой проекта для ПЛИС. В первом случае часто отдается приоритет таким требованиям как кроссплатформенность и кратчайшие сроки разработки проекта, а оптимизация кода требуется только в критических местах, требующих максимального быстродействия. Очень часто в таких случаях используются низкоуровневые «вставки». Процесс создания интегральных схем автоматизирован, описание логики работы и функционала микросхемы происходит так же на специальных языках высокого уровня.

Таким образом, для более эффективного использования ресурсов ПЛИС необходимо использовать низкоуровневую модель, учитывающую избыточность описания. Кроме того, использование низкоуровневой модели позволит получить более полную информацию о часто встречающихся ошибках проектирования, например о критических задержках сигнала в ПЛИС [3]. Особенно актуально это для проектов, в которых ПЛИС используются как системы на кристалле.

Библиографический список

1. Xilinx DS312 Spartan-3E FPGA Family Data Sheet. www.xilinx.com/support/documentation/data_sheet/ds312.pdf

2. Кузелин М.О., Кнышев Д.А., Зотов В.Ю. Современные семейства ПЛИС фирмы Xilinx: Справ. пособие. — М.: Горячая линия-Телеком, 2004. - 440 с.

3. Мальцев Г.Н, Панкратов А.В, Макунин А.А. Анализ структуры исходных файлов проекта // Информационно-управляющие системы. 2014. №6 (73). С. 94 –101.