

УДК 004.3

МЕТОДИКА ПОЛУЧЕНИЯ КА МУРА ИЗ KISS2-СПЕЦИФИКАЦИЙ И ЕЕ ПРИМЕНЕНИЕ К ИССЛЕДОВАНИЯМ В БАЗИСЕ FPGA

Татолов Е.Р., Солдатов К.А., Зеленёва И.Я.

Донецкий национальный технический университет, Украина

Рассмотрены назначение, структура и особенности формата описания конечных автоматов (КА) KISS2, сформулирована методика получения моделей Мура на его основе. Показаны основные идеи применения методики к исследованию эффективности алгоритмов кодирования состояний при синтезе КА Мура для базиса FPGA с использованием HDL-технологий и XST (Xilinx Synthesis Technology).

Введение

Синхронные последовательностные логические схемы (СПЛС) входят в состав цифровых устройств как регулярные элементы (регистры, счетчики) и системно-управляющие узлы [1]. Конечный автомат (КА) является наиболее общей моделью СПЛС и позволяет проводить операции ее синтеза, анализа, композиции, декомпозиции, минимизации, оптимизации на абстрактном и структурном уровнях [2].

Спецификация КА осуществляется с использованием как классических средств (графов и таблиц переходов, граф-схем алгоритмов, ASM-диаграмм) [1, 3], так и компьютерно-ориентированных форматов [4], используемых САПР электронных устройств и системами документирования проектов.

Текстовый формат спецификации KISS2 описывает КА множеством его переходов [5], обеспечивая читабельность и удобство автоматизированной обработки. Широкое распространение формат KISS2 получил благодаря библиотекам тестовых КА (LGSynth89, LGSynth91, LGSynth93) [6], используемым для апробации алгоритмов минимизации и кодирования состояний, подходов к реализации логической схемы. Поскольку оценка эффективности методик синтеза КА зависит не только от стратегии оптимизации (площадь, быстродействие), но и от целевого базиса реализации (ASIC, SPLD, CPLD, FPGA), то необходимыми являются исследования для ряда аппаратных платформ с применением указанных библиотек, что требует адаптации тестовых КА под базисно-ориентированные средства разработки.

Парадигма проектирования для современных цифровых микросхем, основанная на использовании HDL-технологий и комплексных САПР, и необходимость в библиотеках тестовых КА обусловили появление программных средств преобразования (конвертеров) KISS2-файлов в VHDL и Verilog-модели [7]. Однако, указанные конвертеры генерируют HDL-описания КА Мили, не предоставляя возможности получения КА Мура.

В данной работе рассматриваются теоретические и практические аспекты методики получения КА Мура из KISS2-спецификаций, а также вопросы построения их HDL-моделей, ориентированных на средство синтеза XST (Xilinx Synthesis Technology) [8] и поддерживающих возможности управления алгоритмом кодирования состояний.

Формат спецификации КА KISS2

Формат KISS2 является текстовым и определяет КА с помощью декларативного заголовка и списка переходов [5]. Структура заголовка содержит указание количества входов (префикс «i»), выходов (префикс «o») и состояний КА (префикс «s»); размерности списка переходов (префикс «r») и идентификатора начального состояния (префикс «t»):

$$f_k - n_k \quad (1)$$

где f_k – префикс; n_k – числовое значение либо идентификатор (в случае задания начального состояния);

«→» – пробельные символы; k – порядковый номер элемента заголовка.

Список переходов представляет собой набор последовательностей вида

$$x_h - a^m h - a^s h - y_h, \quad (2)$$

где x_h – входная последовательность, инициирующая переход; a^m_h – состояние, из которого осуществляется переход; a^s_h – состояние, куда осуществляется переход; y_h – выходная последовательность, генерируемая при переходе; «→» – пробельные символы; h – порядковый номер перехода в списке переходов.

Последовательности x_h и y_h включают элементы расширенного двоичного алфавита $\{0, 1, -\}$, где символ «→» означает произвольное логическое значение. Состояния a^m_h и a^s_h определяются идентификаторами, допускающими малые и большие латинские буквы, цифры и символы подчеркивания. Кроме того, в качестве состояния a^m_h может использоваться зарезервированное значение «*», что приводит к формированию переходов из каждого состояния КА в состояние a^s_h с последовательностями x_h и y_h . Если в заголовке отсутствует явное указание начального состояния с применением префикса «г», то начальным состоянием КА выступает состояние a^m_h первого перехода, у которого $a^m_h \neq \text{«*»}$.

На рис. 1 показано содержимое файла `mc.kiss2` из библиотеки КА `LGSynth93` [6], где с помощью (1) и (2) задан КА Мили. Поскольку в заголовке отсутствует указание начального состояния, то в его качестве принимается состояние `HG`.

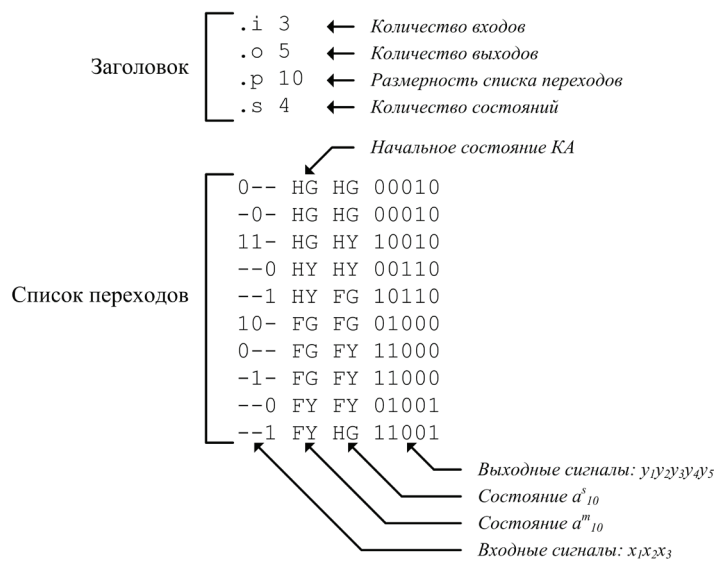


Рисунок 1. Содержимое файла `mc.kiss2`

Методика получения КА Мура

Поскольку стиль описания переходов КА в формате `KISS2` предполагает указание выходных последовательностей y_h (2), то, в общем случае, имеет место модель Мили [1, 3]. Получение соответствующих КА модели Мура возможно с использованием методов абстрактной теории автоматов [2] и сводится к следующему:

1. Анализ синтаксической корректности `KISS2`-файла.
2. Обработка переходов с $a^m_h = \text{«*»}$.
3. Удаление эквивалентных переходов.
4. Анализ семантической корректности `KISS2`-файла.
5. Формализация КА Мили.
6. Формирование множества состояний КА Мура.
7. Расширение множества состояний КА Мура за счет преходящих состояний.
8. Формирование функций переходов и выходов КА Мура.
9. Формализация КА Мура.

На первом этапе осуществляется проверка общей структуры KISS2-файла; соответствия количества входов, выходов, состояний, переходов из заголовка списку переходов; допустимости заданного начального состояния. Затем проводится обработка переходов, у которых $a_h^m = \langle \langle * \rangle \rangle$ – они заменяются множествами переходов. Далее выполняется удаление переходов, имеющих попарно совпадающие компоненты x_h , a_h^m , a_h^s , y_h , и проверка на отсутствие коллизий, связанных с наличием нескольких переходов с одинаковыми последовательностями x_h для одного и того же a_h^m . После этого может быть сформировано формальное описание КА Мили.

Согласно абстрактной теории автоматов, для любого КА Мили может быть получен эквивалентный КА Мура. Процесс построения КА Мура начинается с формирования множества состояний. При наличии проходящих состояний в КА Мили (таких состояний, которые не имеют ни одного входящего перехода), выполняется расширение множества состояний КА Мура. Затем, используя методы из [2], можно получить функции переходов и выходов, а также формализовать представление КА.

На рис. 2 показан граф переходов КА Мура, соответствующего тестовому КА mc.kiss2 и полученному с использованием описанной методики.

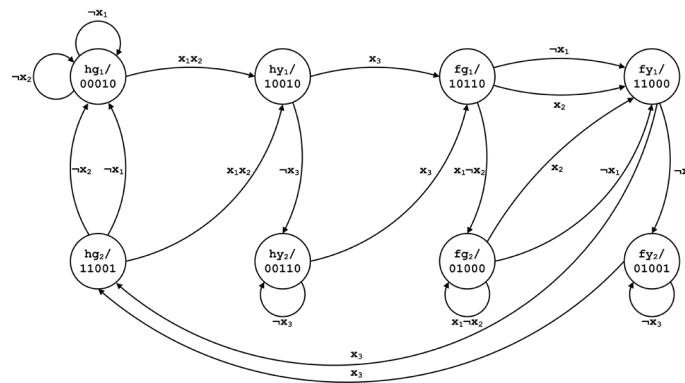


Рисунок 2. Граф переходов КА Мура, соответствующего тестовому КА mc.kiss2

Принципы применения методики к исследованиям в базисе FPGA

Представленная методика может быть использована для исследования эффективности алгоритмов кодирования состояний при синтезе КА Мура в базисе современных микросхем FPGA фирмы Xilinx. Для этого, формализованная модель Мура должна быть преобразована в HDL-описание в соответствии с техниками распознавания КА средства синтеза XST (Xilinx Synthesis Technology) [8]. Кроме того, в HDL-файл должна быть включена информация о способе кодирования состояний, который необходимо применять при синтезе. Это может быть один из автоматически поддерживаемых XST методов (унитарное, компактное, последовательное) или специфическое кодирование, задаваемое непосредственно в HDL-коде.

Соответствующая тестовому КА mc.kiss2 модель Мура (рис. 2) была описана на языке VHDL с использованием двух процессов [8] и синтезирована с помощью XST при различных вариантах кодирования состояний. Результаты исследований для микросхемы XC5VLX30 (семейство Virtex-5), характеризующие количество LUT-ячеек и максимальную частоту функционирования КА, представлены на рис. 3 и 4.

Если целью оптимизации является уменьшение аппаратных затрат, то наиболее оптимальным вариантом выступает компактное кодирование (7 LUT-ячеек), которое требует в 1,86 раза меньше ресурсов, чем унитарное кодирование (13 LUT-ячеек). Однако, именно унитарное кодирование состояний приводит к наиболее быстрой реализации логической схемы КА, обеспечивая максимальную частоту функционирования 1071,123 МГц. Алгоритм кодирования состояний, рекомендуемый XST по умолчанию (кодирование «auto»), для данного КА позволяет получить неоптимальный, но достаточно низкий уровень аппаратных затрат (8 LUT-ячеек), при значительном быстродействии (1058,425 МГц).

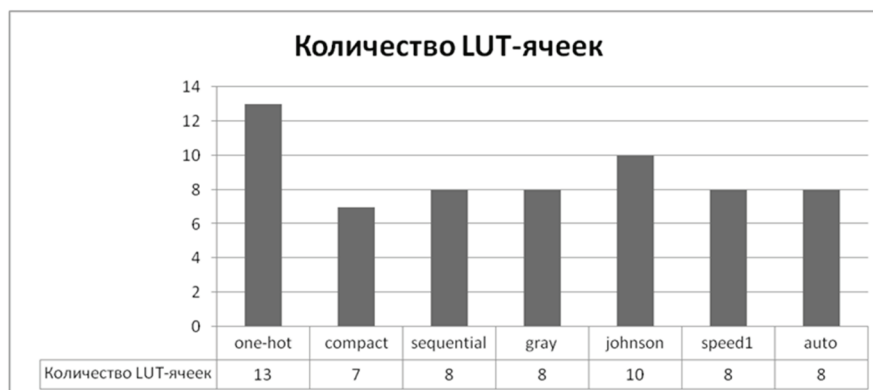


Рисунок 3. Количество необходимых для синтеза КА LUT-ячеек

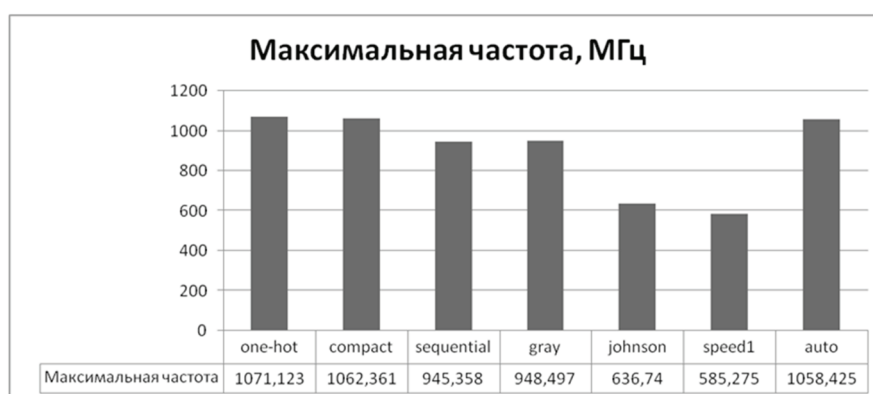


Рисунок 4. Максимальная частота функционирования КА

Выводы

Разработанная методика трансформации KISS2-спецификаций в модели Мура может использоваться для формирования набора тестовых КА на основе известных библиотек [6]. Тестовые КА позволяют проводить исследования эффективности алгоритмов оптимизации для различных аппаратных платформ и стратегий синтеза и должны быть предварительно приведены к базисно-ориентированным средствам разработки (например, HDL-моделям).

Дальнейшие направления исследований связаны с разработкой автоматизированного конвертера KISS2-файлов в КА Мура и их VHDL-модели, ориентированные на средство синтеза XST и поддерживающие возможности задания алгоритма кодирования состояний, способа реализации логической схемы, общей стратегии оптимизации. С помощью конвертера планируется осуществить комплексное исследование особенностей реализации моделей Мура в базе FPGA-микросхем фирмы Xilinx.

Литература

- [1] Баркалов А.А. Синтез микропрограммных автоматов на заказных и программируемых СБИС / А.А. Баркалов, Л.А. Титаренко. – Донецк: ДонНТУ, Технопарк ДонНТУ УНИТЕХ, 2009. – 336 с.
- [2] Баранов С.И. Синтез микропрограммных автоматов (граф-схемы и автоматы) / С.И. Баранов. – Л.: Энергия, 1979. – 232 с.
- [3] Nelson V. Digital logic circuit analysis and design / V. Nelson, H. Nagle, J. Irwin, B. Carroll. – Prentice Hall, 1995. – 842 pp.
- [4] Perkowski M. Digital design automation – finite state machine design. Электронный ресурс. –

Режим доступа: <http://web.cecs.pdx.edu/~mperkows/PerkowskiGoogle/finite-sm.pdf>.

- [5] Yang S. Logic synthesis and optimization benchmarks user guide. Version 3.0. Technical Report / S. Yang. – North Carolina, 1991. – 44 pp.
- [6] International workshop on logic synthesis benchmark suite (LGSynth93). Электронный ресурс. – Режим доступа: <http://www-lab13.kuee.kyoto-u.ac.jp/eda/benchmark/mcnc/benchmarks/LGSynth93/LGSynth93.tar>.
- [7] Abdel-Hamid A. A tool for converting finite state machine to VHDL / A. Abdel-Hamid, M. Zaki, S. Tahar // Proceedings of IEEE Canadian conference on electrical and computer engineering. – Ontario, 2004. – pp. 1907-1910.
- [8] Xilinx Inc. XST User Guide. – 2009. – 485 pp.