

СИСТЕМА АВТОМАТИЧЕСКОГО ПРОЕКТИРОВАНИЯ В СРЕДЕ А-HDL УСТРОЙСТВ, ПРЕДСТАВЛЕННЫХ КОНЕЧНЫМИ АВТОМАТАМИ

Бережок А.Ю., Зеленева И.Я.
Донецкий национальный технический университет

Микросхемы, программируемые пользователем, открыли новую страницу в истории современной микроэлектроники и вычислительной техники. Вследствие этого БИС/СБИС, предназначенные для решения специализированных задач, стандартной продукцией электронной промышленности со всеми вытекающими из этого положительными последствиями: массовое производство, снижение стоимости микросхем, сроков разработки и выхода на рынок продукции на их основе.

В данный момент существует множество классов ПЛИС, но наиболее часто используются следующие ИС: SPLD, CPLD, FPGA. [3]

В настоящее время одной из актуальных тем является тема конечных автоматов. Конечный автомат представляет собой граф состояний и переходов, который описывает, каким образом описываемый объект реагирует на получение событий. [2]

Связующим звеном между конечным автоматом и аппаратной реализацией автомата является ПЛИС.

Active-HDL в своей программной реализации имеет средства для создания и редактирования FSM-структур (finite state machine, конечный автомат). Основная идея автоматной реализации – это описание процесса в виде граф-схемы с генерацией программного кода на языке VHDL или Verilog, с последующей возможностью синтеза полученного кода на заданной ПЛИС.

В настоящее время рост использования ПЛИС для реализации автоматов приостанавливается дороговизной микросхем. Поэтому конструкторам-программистам приходится решать задачу уменьшения аппаратных затрат (в данном случае – площадь кристалла ПЛИС или число вентилях) с большими затратами времени разработки.

Существует множество методов уменьшить аппаратные затраты путем использования дополнительных элементов (например мультиплексоров, ППЗУ), которые придают алгоритму гибкость автомата с программируемой логикой (АПЛ) и скорость автомата с жесткой логикой (АЖЛ).

Сравнительные результаты параметров синтезированных устройств при одинаковых исходных граф-схемах, реализованные средствами FSM А-HDL (**finite state machine**) и посредством структурного описания результирующей схемы, показывают определенное уменьшение числа используемых элементов FPGA при реализации вторым способом.

Результаты синтеза (для ПЛИС Lattice ispGDX2 Part LX256V):

Таблица 1 – Результат синтеза в FSM редакторе

DFF	8 uses
IBUF	2 uses
OBUF	3 uses
AND2	16 uses
INV	12 uses

Таблица 2 – Результат синтеза тривиальным методом

DFF	6 uses
IBUF	2 uses
OBUF	3 uses
AND2	12 uses
INV	11 uses

Недостаток FSM-редактора А-HDL – генерация программного кода в поведенческом стиле, что делает невозможным предсказать результат синтеза.

Структурное же описание схемы, полученной в результате тривиальной реализации, дает возможность предсказания результатов еще до синтеза – на этапе генерации программного кода, что дает возможность генерировать код для кристалла или числа вентилях FPGA заданной размерности.

При реализации схемы на SPLD прогнозируемая площадь кристалла рассчитывается по формуле: $S = 2*(L+R)*H + H*(N+R)$, где S-площадь кристалла ПЛИС, L-число логических условий, N-число выходных функций графа, R-разрядность кода состояния, H-число всех возможных переходов автомата. [1]

При реализации схемы на FPGA анализ сводится к следующим формулам:

1. число D-триггеров используемых в схеме: $K=N+R$;
2. максимальное число инверторов схемы: $I=M+R$, где M-количество состояний; минимальное же число можно определить путем склейки термов формул для Q_i и D_i (где Q_i – формула для i-ого состояния, D_i – формула для i-ого разряда кода состояния);
3. Максимальное число элементов 2И: $A_i = \sum_{j=1}^F (T - 1) + (F - 1)$, где T-число составляющих j-ого терма i-й формулы, F-число термов i-й формулы, минимальное же число определяется после минимизации формул, аналогичным образом.
4. Данные формулы рассмотрены для FPGA с инвертором, 2И-элементом и D-триггером в узле (Xilinx CoolRunner XPLA3, Cypress Delta39K и т.д.).

На основании выполненного анализа и сравнения возникает необходимость программной реализации методов генерации структурного кода граф-автоматов и применение оптимизации этого кода.

В ходе анализа методов оптимизации и постоянным расширением их числа появляются определенные требования к реализации программного продукта, формируется определенная структура.



Рис. 1. Структура разрабатываемой САПР

Разрабатываемая система состоит из следующих частей:

- 1) Графический редактор, который реализует возможность графического отображения, редактирования и создания требуемой граф-схемы (FSM).
- 2) Компилятор, который дает возможность текстуального набора графа и приведение текстуального представления во внутренний язык программы.
- 3) Генератор кода с различными оптимизаторами, которые преобразуют внутренний код графа в конкретную (Мура или Мили) реализацию автомата.

Разрабатываемая программа производит оптимизацию выбранным методом и осуществляет анализ полученной структуры по вышеописанным методам.

Литература

- [1] А.А. Баркалов Синтез устройств управления на программируемых логических устройствах. – Донецк, 2002. – 262 с.
- [2] Ю.Г. Карпов Теория автоматов. – С-Пб., Питер, 2002. – 206 с.
- [3] Е.А. Суворова, Ю.Е. Шейнин Проектирование цифровых систем на VHDL. – С-Пб., БХВ-Петербург, 2003. – 560 с.