ИСПОЛЬЗОВАНИЕ СВОЙСТВ ЦИФРОВЫХ АВТОМАТОВ ДЛЯ ОБФУСКАЦИИ И ВНЕДРЕНИЯ ВОДЯНЫХ ЗНАКОВ В ОПИСАНИЯ ЦИФРОВЫХ УСТРОЙСТВ

В.В. Сергейчик , А.А. Иванюк
Кафедра информатики, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: vovasq@mail.ru, ivaniuk@bsuir.by

Рассматривается лексическая и функциональная обфускация HDL-описаний цифровых устройств с использованием свойств эквивалентных состояний полностью определенных конечных автоматов. Предлагаются методы обфускации и внедрения водяных знаков.

Введение

Обфускация представляет собой запутывание понимания функционирования или структуры схемы с целью защиты от обратного проектирования. Обфускация может затрагивать различные уровни абстракции проектных описаний. В случае HDL-языков выделяют две разновидности обфускации: лексическую и функциональную (схемную) [1]. Лексическая затрагивает только уровень исходного кода, не внося изменений в результат синтеза. Функциональная обфускация приводит к усложнению не только исходного кода, но и результирующей схемы. Идея водяных знаков заключается во внедрении в описание устройства секретного сообщения, наличие которого в случае судебного разбирательства позволит доказать авторство. Обфускацию часто применяют совместно с водяными знаками для усложнения их удаления или модификации.

I. Существующие подходы

Подходы к лексической обфускации подробно описаны в [2]. Примерами схемной обфускации могут быть: схемы с «заиканием» [3], введение комбинационных циклов с использованием простых логических импликаций, расширение пространства состояний конечных автоматов (КА) [4]. Применительно к КА необходимо упомянуть подходы внедрения водяных знаков в выходные значения [5], кодировку состояний, топологию КА.

II. Эквивалентные состояния

Математически КА описывается набором из шести объектов: конечное множество входных символов (входной алфавит) Σ; конечное множество выходных символов (выходной алфавит) Δ; конечное множество состояний $Q = \{q_0, q_1, \ldots, q_{r-1}\}$; начальное состояние $q_0 \in Q$; функция перехода из одного состояния в другое $\delta(q, x)$; функция выходов $\lambda(q, x)$. КА называют полностью определенным, если функции δ, λ определены для всех пар $(q,\ x) \in Q \times \Sigma$ [6].

В теории синтеза полностью определенных КА существует понятие избыточных или эквивалентных состояний. Два состояния, q_i в КА P и q_j в КА R (при этом R может быть копией P), называются *эквивалентными*, если для P, изначально находившегося в q_i, и R в q_j, не существует последовательности входных символов, которая привела бы к появлению на выходах P и R различающихся выходных символов [7]. Рассмотрим два состояния s и t КА на рис. 1а, последовательности выходных символов $S = (y_0, y_1, \ldots, y_{m-1})$, $y_k \in \Delta$ для всех возможных последовательностей из m входных символов $T = (x_0, x_1, \ldots, x_{m-1})$, где $m = |Q| = 4$, $x_k \in \Sigma$ приведены в таблице 1. По таблице легко убедиться, что состояния s и t эквивалентны.

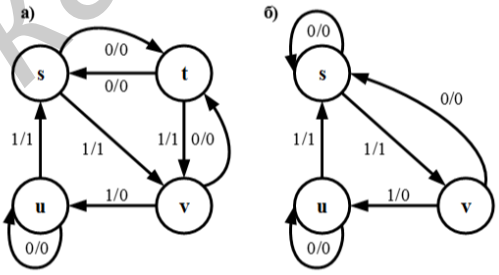


Рис. 1 – а) КА с эквивалентными состояниями; б) минимизированный КА

Таблица 1 – Выходные значения для s и t

T	S_s	S_t	T	S_s	S_t
0000	0000	0000	1000	1000	1000
0001	0001	0001	1001	1001	1001
0010	0010	0010	1010	1010	1010
0011	0010	0010	1011	1010	1010
0100	0100	0100	1100	1000	1000
0101	0101	0101	1101	1001	1001
0110	0100	0100	1110	1010	1010
0111	0101	0101	1111	1011	1011

Сокращенный КА представляет собой минимальную по числу состояний версию КА для реализации заданной функциональности, он не содержит избыточности [7]. Для получения сокращенного КА необходимо оставить ровно по одному состоянию-представителю каждого класса эквивалентности, например, на рис. 1б из двух эквивалентных состояний s и t было оставлено состояние s.

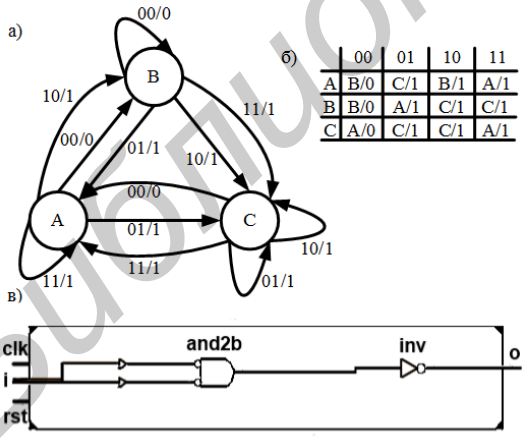
Экспериментом длины k называют подачу на входы КА, находящегося в заданном начальном состоянии, последовательности из k входных символов. Два состояния *неразличимы* для любого эксперимента длины k, если для всех экспериментов длины k результат не зависит от того, какое из этих состояний было начальным [7].

III. Обфускация с использованием эквивалентных состояний

Введение эквивалентных состояний можно использовать для обфускации. Некоторые средства синтеза проводят минимизацию числа состояний КА. В таком случае будет иметь место лексическая обфускация: дополнительные состояния окажутся в итоге минимизированы. Однако при изучении исходного кода злоумышленнику придется исследовать избыточный граф передачи состояний (ГПС). Простым способом создания эквивалентных состояний может быть расщепление узлов ГПС, ссылающихся на себя.

Эксперименты показывают, что некоторые средства синтеза не проводят подобную минимизацию. Например, xilinx ise 12, synplify pro 2010 не обнаруживают эквивалентные состояния. В этом случае, введение подобных состояний можно использовать для схемной обфускации.

Возможна ситуация, когда все состояния окажутся попарно-эквивалентными, тогда конечный автомат не будет проявлять последовательностного поведения, а будет вести себя, как комбинационная схема. Пример КА, реализующего логическое ИЛИ, показан на рис. 2а. Результат синтеза приведен на рис. 2в. Такие вырожденные случаи обнаруживаются и минимизируются исследованными средствами синтеза.



	00	01	10	11
A	B/0	C/1	B/1	A/1
B	B/0	A/1	C/1	C/1
C	A/0	C/1	C/1	A/1

Рис. 2 – а) КА с попарно эквивалентными состояниями; б) таблица переходов и выходов КА; в) результат синтеза

Для описания произвольной комбинационной схемы C с помощью КА нужно в каждой строке таблицы переходов и выходов в качестве выходных значений использовать соответствующие выходные значения таблицы истинности C, следующие состояния можно выбирать произвольно.

IV. Внедрение водяных знаков

Избыточность, вносимую эквивалентными состояниями можно использовать для постановки водяных знаков. В случае комбинационной схемы, описанной с помощью КА, водяной знак можно закодировать в следующих состояниях, назначив каждому из них двоичный код и подставляя в таблице переходов и выходов состояние с кодом, соответствующим порции бит водяного знака. Следует отметить, что такой водяной знак не сохраняется в ходе синтеза, поэтому подход применим лишь для защиты исходных HDL-описаний. Для последовательностной схемы биты водяного знака можно скрыть, пронумеровав состояния с циклами и расщепив на эквивалентные те из них, номера которых соответствуют единичным битам в водяном знаке.

Неразличимые для любого эксперимента длины k состояния можно использовать для внедрения водяного знака. Для этого выбирается $k < |Q|$, затем из Q выбираются j состояний (где j – число бит водяного знака), принадлежащих классам эквивалентности меньше k. Далее возможны вариации: дополнять каждое состояние до класса эквивалентности с номером k (для бита «0» водяного знака) или $k+1$. Второй вариант: каждое состояние дополняется до класса эквивалентности k путем добавления новых переходов (и, возможно, новых входных переменных). Выходное значение для $(k+1)$ -го перехода равно соответствующему фрагменту водяного знака. При извлечении осуществляется поиск всех состояний k-го класса эквивалентности, а затем определяются значения последнего перехода. Добавление переходов приводит к усложнению понимания схемы и может увеличить аппаратные издержки. k-эквивалентные состояния не минимизируются, поэтому водяной знак будет присутствовать в схеме после синтеза.

1. Иванюк, А.А. Проектирование встраиваемых цифровых устройств и систем: монография / А. А. Иванюк. — Минск: Бестпринт, 2012. — 337с.
2. Collberg, K. A taxonomy of obfuscating transformations / C. Collberg, C. Thomborson, D. Low; Techinical Report 148. – Auckland, New Zealand. – 1997. – 36 p.
3. Li, L. Structural transformation for best-possible obfuscation of sequential circuits / L. Li, H. Zhou // Hardware Oriented Security and Trust (HOST), 2 – 3 June 2013 / Austin, USA – 2013. — Pp. 55 — 60.
4. Chakraborty, R.S. Hardware Protection Through Design Obfuscation: PhD thesis / R. S. Charaborty. – Cleveland, USA: 2010 – 183 p.
5. Abdel-Hamid, A.A Survey on IP Watermarking Techniques / A. Abdel-Hamid, Sofiene Tahar, El Mostapha Aboulhamid // Design Automation for Embedded Systems. — 2004. — Vol. 9. — Pp. 211 — 227.
6. Савельев, А.Я. Прикладная теория цифровых автоматов: учебник для вузов / А. Я. Савельев. – М.: Высшая Школа, 1987. – 272 с.
7. Mealy, G.H. A Method for Synthesizing Sequential Circuits / G. H. Mealy // The Bell System Technical Journal. – 1955. – September. – Pp. 1045 – 1078.