УДК 681.32 DOI: 10.31114/2078-7707-2022-3-10-17

# Обратное проектирование СБИС для обеспечения безопасности аппаратуры

Д.И. Черемисинов, Л.Д. Черемисинова

Объединенный институт проблем информатики НАН Беларуси, г. Минск

{cher, cld}@newman.bas-net.by

*Аннотация* — **Обратное проектирование СБИС – мощный инструмент, используемый при верификации проекта для повышения быстродействия средств моделирования аппаратуры, а также для обнаружения незаконных вложений в процессе ее производства. Задача обратного проектирования СБИС заключается в построении спецификации устройства путем анализа его аппаратной реализации в виде СБИС. Основным этапом обратного проектирования является декомпиляция плоского нетлиста транзисторной схемы, которая состоит в извлечении из него описания на уровне логических элементов. В работе предлагаются программные средства выделения логической схемы из плоского SPICE-описания транзисторной схемы. Приводятся примеры обратного инжиниринга практических примеров транзисторных схем.**

*Ключевые слова* — **КМОП схемы из транзисторов, экстракция транзисторных подсхем, распознавание логических вентилей, формат SPICE.**

## I. Введение

Микроэлектронные системы в настоящее время строятся обычно из готовых компонентов, приобретаемых через глобально распределенную ненадежную цепочку поставок. Отсутствие доверия к этим компонентам требует дополнительной проверки компонентов перед использованием [1], [2]. Кроме того, серьезной проблемой становятся аппаратные трояны [3]. Интегральную схему можно считать подлинной, когда она либо изготовлена на надежном предприятии, либо её характеристики проверены с помощью полномасштабного обратного проектирования [4]. Мощным и надежным инструментом для обеспечения безопасности аппаратуры является обратное проектирование или обратный инжиниринг СБИС (hardware reverse engineering), который позволяет обнаруживать несанкционированные вкладки и одновременно может использоваться для перепроектирования и верификации интегральных схем. Задача обратной инженерии инверсна задаче проектирования СБИС в смысле направления процесса преобразований, она заключается в построении спецификации устройства путем анализа его аппаратной реализации в виде СБИС.

Чтобы сделать вывод о высокоуровневой функциональности неструктурированного списка соединений транзисторов исследуемой СБИС, необходимо определить границы вентилей для последующего анализа их иерархии. Обратный инжиниринг с уровня плоского списка соединений транзисторов исследуемой СБИС представляет собой решение двух ключевых технических проблем: 1) восстановление границ вентилей и иерархии их соединений по списку соединений транзисторов и 2) сопоставление полученных вентилей с известными компонентами библиотеки. Основным этапом обратного проектирования является декомпиляция плоской транзисторной схемы, которая состоит в извлечении описания уровня логических элементов.

В работе предложены методы и программные средства для решения ключевых задач, возникающих на этапе декомпиляции транзисторных схем: разбиение графа на компоненты связности, соответствующие транзисторным подсхемам; распознавание подсхем, являющихся логическими элементами, и реализуемых ими функций; распознавание топологически эквивалентных транзисторных подсхем; формирование библиотеки распознанных элементов и построение сначала двухуровневой транзисторной схемы, а затем логической сети. Рассматривается наиболее распространенный стиль логики – логические комплементарные МОП-структуры.

Предполагается, что декомпилированная схема может кроме распознаваемых КМОП элементов и передаточных логических элементов (pass gates) может содержать также и другие структуры, выделяемые при декомпиляции как псевдоэлементы (не КМОП вентили), а также отдельные транзисторы. После формирования библиотеки распознанных элементов и построения двухуровневой транзисторной схемы формируется логическая сеть: определяются входы и выходы логической сети как блока декомпилируемой схемы, находится ее функциональное описание и представление на языке SF иерархических структурно-функциональных описаний дискретных устройств [5], который является внутренним языком программных средств проектирования компонентов СБИС [6] в Объединенном институте проблем информатики НАН Беларуси. Возможна трансляция полученного SF-описания на языки VHDL и Verilog.

## II. Декомпиляция транзисторных схем

Исходным объектом при декомпиляции является плоское описание схемы исследуемой СБИС, которое

состоит из *p*-МОП- и *n*-МОП-транзисторов. Для этой схемы может иметься заданная априори библиотека логических элементов, использованная при ее проектировании. В этом случае обратный инжиниринг с уровня плоского описания схемы транзисторов состоит в решении двух ключевых проблем: 1) выделение правильных транзисторных подсхем, которые выглядят как логические элементы, и иерархии их соединений и 2) распознавание тех из подсхем, которые реализуют логические элементы, путем сравнения их со схемами библиотечных элементов. Распознавание логических элементов осуществляется либо на функциональном уровне путем сравнения реализуемых подсхемами функций, либо на структурном уровне путем анализа изоморфизма графов соединений транзисторов.

Предложенный в [7] метод декомпиляции извлекает структуру функционального уровня из схемы транзисторного уровня для наиболее распространенного стиля логики – комплементарных МОП-структур, относящихся к классу статических схем, в которых в каждый момент времени выход элемента соединяется либо с цепью питания, либо с землей через тракт с малым сопротивлением. Рассматривается случай, когда библиотека логических элементов не известна и строится в процессе декомпиляции транзисторной схемы.

Метод реализует трехэтапный процесс. Сначала, используется структурный подход [8] к декомпиляции транзисторных схем, который позволяет разбить транзисторную схему на непересекающиеся подсхемы, представляющие группы транзисторов, связанных по постоянному току. Группой транзисторов, связанных по постоянному току, является произвольная схема из МОП-транзисторов с тремя типами внешних соединений: 1) входы группы подаются только на затворы транзисторов группы; 2) выходы группы подаются только на затворы транзисторов других групп; 3) имеются связи транзисторов группы с шинами питания *Vdd* и земли *Gnd*. Например, схема, приведенная на рис. 1, содержит две такие группы, выделенные пунктирной линией.

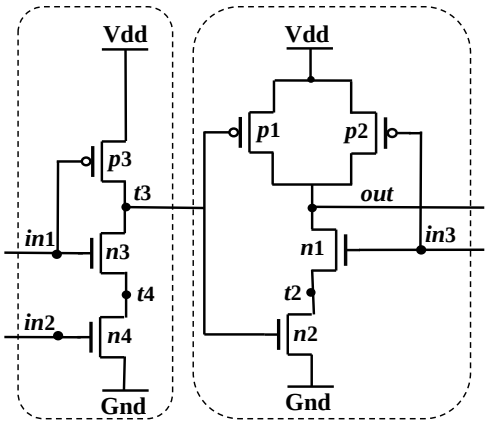


**Рис. 1. Выделение групп транзисторов, связанных по току**

Среди найденных групп транзисторов выделяются правильные подсхемы, которые представляют собой статические КМОП-вентили, и определяются реализуемые ими функции. Группа транзисторов, связанных по току, реализует КМОП-вентиль, если удовлетворяет следующим условиям: 1) транзисторы *p*-блока расположены между цепью *Vdd* и цепью выхода группы, а транзисторы *n*-блока – между цепями выхода и *Gnd*; 2) все пути из цепи выхода доходят до цепей питания (*Gnd* или *Vdd*), и наоборот; 3) *p*- и *n*-блоки группы имеют одинаковое количество транзисторов и реализуют взаимно инверсные функции. Например, правая из двух выделенных на рис. 1 групп транзисторов, связанных по току, удовлетворяет этим условиям и реализует вентиль И-НЕ, а левая – нет. Каждая из подсхем, не распознанных как КМОП-вентиль, объявляется нераспознанным псевдоэлементом. Функциональное описание такого элемента не известно.

В работе [8] рассматриваются задачи и графовые методы их решения, которые возникают при структурном поиске в транзисторной схеме групп связанных по постоянному току транзисторов, распознавании подсхем, представляющих КМОП-вентили, а также передаточных элементов (пар транзисторов n-МОП и p-МОП типа, соединённых параллельно выводами стока и истока). Предлагается также и метод построения канонического представления функций, реализуемых распознанными вентилями. Это позволяет разбить множество найденных КМОП-вентилей на подмножества функционально эквивалентных. Множество псевдоэлементов также разбивается на подмножества (для дальнейшего анализа) элементов, имеющих одинаковое число транзисторов и их связей.

Некоторые особенности топологической реализации схем на транзисторном уровне приводят к существованию подсхем, реализующих одну и ту же логическую функцию, но различающихся в топологическом плане. В работе [9] рассматривается графовая постановка и метод решения задачи распознавания топологически эквивалентных транзисторных подсхем. На этом этапе имеется множество связных помеченных подграфов, соответствующих транзисторным подсхемам, найденным структурным методом. Задача сводится к разбиению множества таких подграфов на классы изоморфных графов, задающих топологически эквивалентные подсхемы (вентилей и псевдоэлементов). Представители этих классов будут задавать найденные библиотечные элементы, которые формируют второй уровень описания транзисторной схемы.

## III. Задание транзисторных схем

Исходные транзисторные схемы представляются в формате SPICE (Simulation Program with Integrated Circuit Emphasis) для обмена электрическими схемами [10]. В этом формате электрические схемы состоят из элементов, которые соединены друг с другом цепями. Главной частью описания схемы в формате SPICE является список транзисторов, в котором для каждого вывода транзистора (сток, затвор, исток, подложка) указано имя цепи, соединяющей его с остальными частями схемы. Общая форма описания связей униполярного транзистора в формате SPICE имеет вид:

<name> <nd> <ng> <ns> <nb> <model-name>,

где name – название транзистора; nd, ng, ns и nb – идентификаторы цепей, связанных с выводами стока (drain), затвора (gate), истока (source) и подложки (substrate) соответственно, «model-name» – тип транзистора: n-МОП или p-МОП (nmos или pmos). Например, транзисторная схема (рис. 1) в формате SPICE представлена на листинге 1.

**Листинг 1. SPICE-описание схемы из транзисторов**

```
.GLOBAL  Gnd  Vdd
.SUBCKT GG0 in1 in2 in3 out
Mn1 out in3 t2 Gnd nmos
Mn2 t2 t3 Gnd Gnd nmos
Mn3 t3 in1 t4 Gnd nmos
Mn4 t4 in2 Gnd Gnd nmos
Mp1 out t3 Vdd Vdd pmos
Mp2 out in3 Vdd Vdd pmos
Mp3 t3 in1 Vdd Vdd pmos
.ENDS
Circuit GG0 contains 7 device instances.
Class: pmos            instances:   3
Class: nmos            instances:   4
Circuit contains 9 nets.
```

Результатом декомпиляции транзисторной схемы на рис. 1 (листинг 1) является двухуровневая схема, представляемая иерархическим SPICE-описанием (листинг 2), в которое включены модели всех идентифицированных как элементы групп транзисторов, включая не только КМОП-вентили, но и псевдоэлементы (для которых не распознаны реализуемые ими логические функции). Эти модели и составляют извлеченную при декомпиляции библиотеку вентилей. Для каждого вентиля приведены имена входных и выходного полюсов, а также реализуемая функция (точнее, ее инверсия). Для псевдоэлемента указывается число его транзисторов и цепей.

**Листинг 2. Результат декомпиляции и SPICE-описание двухуровневой транзисторной схемы**

```
Circuit GG0 contains 7 device instances.
Class: pmos            instances:   3
  Class: nmos          instances:   4
Circuit contains 9 nets.

Connected Componens = 2
Valid Components = 1
Pass fets = 0
Psevdo Componens = 1 nets =4
Unclassified fets = 0 nets = 0

 (A AND B)       1

Psevdo
(3)(4)  1
Defining cell: GG0_gen
   Defining global node: Gnd
   Defining global node: Vdd
Start of Computation: 15h07m55s 27/08/2021
End of Computation: 15h07m55s 27/08/2021
Computation Time (s): 0.0050

.SUBCKT G0 A B Y
* (A AND B)
M1 Y A 2 Gnd nmos
M2 2 B Gnd Gnd nmos
M3 Y B Vdd Vdd pmos
M4 Y A Vdd Vdd pmos
.ENDS
.SUBCKT P0_0 P0 P1 P2
* (3)(4)
M1 P0 P1 2 Gnd nmos
M2 2 P2 Gnd Gnd nmos
M3 P0 P1 Vdd Vdd pmos
.ENDS
.SUBCKT GG0_gen in1 in2 in3 out
XM0I1 in3 t3 out G0
Fets=nmosn1+nmosn2+pmosp1+pmosp2
t3 in1 in2 P0_0 Fets=nmosn3+nmosn4+pmosp3
.ENDS
```

Описание схемы в целом приведено в конце листинга 2 после описания библиотеки распознанных подсхем: приводятся экземпляры XM**0**I1, XM**0**I2,… и XMP0I1, XMP0I2, … распознанных подсхем вентилей G и псевдоэлементов P (столько раз, сколько они встречаются в схеме, в приведенной схеме по одному разу). Для каждого экземпляра приведены списки фактических идентификаторов входов и выходов, а также номера входящих в них транзисторов исходной схемы.

## IV. Извлечение логической сети из двухуровневой транзисторной схемы

Транзисторная схема в формате SPICE представляется помеченным неориентированным двудольным графом $G = (V_1, V_2, E)$, $V_1 \cap V_2 = \varnothing$. Вершины из $V_1$ соответствуют входам и выходам схемы, выводам транзисторов, группам, составляющих выделенные элементы. Вершины из $V_2$ соответствуют цепям. Каждая из дуг $e \in E$ связывает вершины из разных множеств $V_1$ и $V_2$.

Логическая сеть дискретного устройства отражает его внутреннее строение с точностью до функций, реализуемых его элементами. В графовой интерпретации сетью называется помеченный ориентированный граф $H = (W, A)$, где множество $W$ разбито на три подмножества вершин, соответствующих входам, выходам сети и элементам. Каждая вершина из первых двух подмножеств помечена входной или выходной переменной сети. Вершины из третьего подмножеств помечены функциями, реализуемыми элементами сети

Ориентированный граф $H = (W, A)$ логической сети строится, исходя из неориентированного двудольного графа $G = (V_1, V_2, E)$, соответствующего объектной двухуровневой транзисторной схеме, путем извлечения из него подграфа, включающего только те вершины из $V_1$, которые соответствуют логическим вентилям. Так как помимо таких вершин в множестве $V_1$ могут быть и другие вершины, то извлекаемая логическая сеть в общем случае может описываться несколькими непересекающимися связными графами $H = (W, A)$. Поиск компонент связности $H^* = (W, A^*)$ в графе $G = (V_1, V_2, E)$ осуществляется в процессе его обхода по входящим и исходящим путям от вершин, помеченных как элементы (передаточные или КМОП вентили). Метод поиска позволяет не только найти компоненту связности $H^* = (W, A^*)$, но и получить лексикографическое упорядочение ее вершин, учитывающее достижимость вершин друг из друга, и соответственно ранжировать граф по уровням.

Следующей задачей, связанной с выделением логической сети, является определение ее входных и выходных полюсов. Эта задача решается путем рассмот-

рения полуокрестностей исхода $\Gamma^{+}v$ и захода $\Gamma^{-}v$ для всех вершин $v$ графа $H = (W, A)$. Предлагаемый метод позволяет выделить логическую сеть, ранжируемую лексикографически. От нее производится переход к логическим уравнениям, задающим функции, реализуемые на ее выходных полюсах. Строится функциональное описание логической сети на языке SF [5].

## V. Пример 1 декомпиляции транзисторной схемы

В качестве примера рассмотрим декомпиляцию транзисторной схемы, представляющей полный одноразрядный сумматор из обзора [11] (рис. 2). SPICE-описание этой схемы приведено на листинге 3.

**Листинг 3. SPICE-описание полного одноразрядного сумматора**

```
.SUBCKT adder a b cin cout sum vdd gnd
M0 vdd a 1 vdd p
M1 vdd b 1 vdd p
M2 1 b 2 vdd p
M3 2 a 3 vdd p
M4 1 cin 3 vdd p
M5 3 a 4 gnd n
M6 4 b gnd gnd n
M7 3 cin 5 gnd n
M8 5 a gnd gnd n
M9 5 b gnd gnd n
M10 cout 3 gnd gnd n
M11 vdd 3 cout vdd p
M12 vdd a 6 vdd p
M13 vdd b 6 vdd p
M14 vdd cin 6 vdd p
M15 6 a 7 vdd p
M16 7 b 8 vdd p
M17 8 cin 9 vdd p
M18 6 3 9 vdd p
M19 vdd 9 sum vdd p
M20 9 a 10 gnd n
M21 10 b 11 gnd n
M22 11 cin gnd gnd n
M23 12 a gnd gnd n
M24 12 b gnd gnd n
M25 12 cin gnd gnd n
M26 9 3 12 gnd n
M27 sum 9 gnd gnd n
.ENDS
```

### A. *Декомпиляция плоского описания транзисторной схемы*

В результате декомпиляции транзисторной схемы передаточных элементов не найдено, но обнаружено четыре группы транзисторов, связанных по постоянному току (листинг 4). Каждая из этих групп транзисторов представляет собой схему логического вентиля (для них распознаны реализуемые ими логические функции), соответственно псевдоэлементов нет.

**Листинг 4. Результат декомпиляции**

```
Contents of circuit adder.sp:  Circuit: 'adder'
   Circuit adder contains 28 device instances.
     Class: n            instances:  14
     Class: p            instances:  14
   Circuit contains 19 nets.

   Connected Componens = 4
   Valid Componens = 4
   Pass fets = 0
   Psevdo Componens = 0 nets =0
   Unclassified fets = 0 nets = 0

   ((A AND B AND C) OR (G AND (D OR E OR F)))  1
   ((A AND B) OR (C AND (D OR E)))             1
   A        2
```

Два вентиля из четырех, обнаруженных в результате декомпиляции, являются инверторами, два остальных реализуют функции:

$$\overline{ABC \vee G(D \vee E \vee E)} \text{ и } \overline{AB \vee C(D \vee E)}.$$

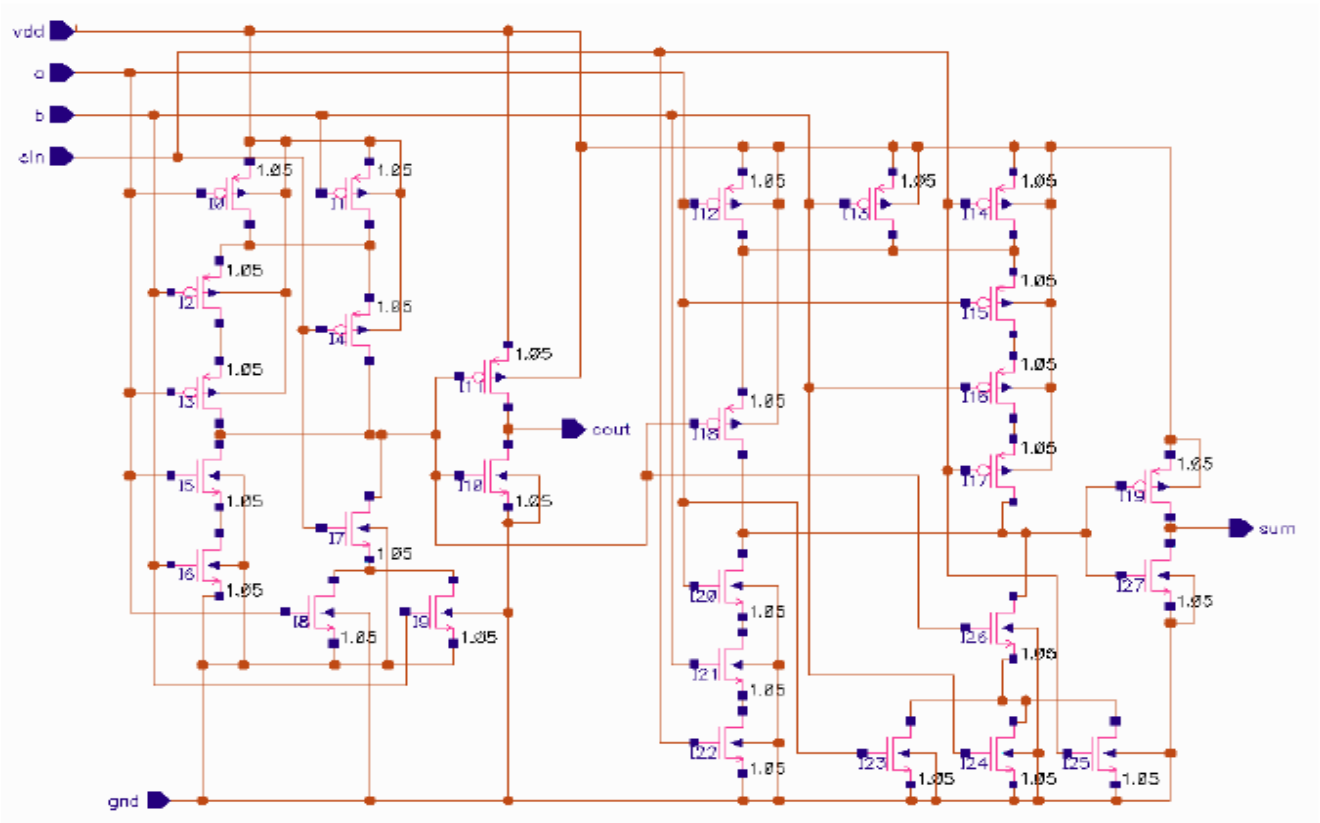


**Рис. 2. Транзисторная схема полного одноразрядного сумматора**

Результатом декомпиляции является двухуровневая транзисторная схема, представляемая иерархическим SPICE-описанием (листинг 5), в которое включены модели трех распознанных подсхем G0, G1 и G2, идентифицированных как КМОП-вентили. Эти модели и составляют извлеченную при декомпиляции библиотеку вентилей.

**Листинг 5. SPICE-описание двухуровневой транзисторной схемы полного одноразрядного сумматора**

```
* SPICE deck for cell adder_gen
.GLOBAL  vdd  gnd
.SUBCKT G0 A B C D E F G Y
* ((A AND B AND C) OR (G AND (D OR E OR F)))
M1 Y A 2 gnd n
M2 2 B 3 gnd n
M3 3 C gnd gnd n
M4 5 D gnd gnd n
M5 5 E gnd gnd n
M6 5 F gnd gnd n
M7 Y G 5 gnd n
M8 vdd A 7 vdd p
M9 vdd B 7 vdd p
M10 vdd C 7 vdd p
M11 7 A 8 vdd p
M12 8 E 9 vdd p
M13 9 F Y vdd p
M14 7 G Y vdd p
.ENDS
.SUBCKT G1 A B C D E Y
* ((A AND B) OR (C AND (D OR E)))
M1 Y A 2 gnd n
M2 2 B gnd gnd n
M3 Y C 4 gnd n
M4 4 D gnd gnd n
M5 4 E gnd gnd n
M6 vdd A 6 vdd p
M7 vdd B 6 vdd p
M8 6 E 7 vdd p
M9 7 A Y vdd p
M10 6 C Y vdd p
.ENDS
.SUBCKT G2 A Y
* A
M1 Y A gnd gnd n
M2 vdd A Y vdd p
.ENDS
.SUBCKT adder_gen a b cin cout sum
XM0I1 a b cin a b cin 3 9 G0
Fets=n20+n21+n22+n23+n24+n25+n26+p12+p13+p14+
p15+p16+p17+p18
XM1I1 a b cin a b 3 G1
Fets=n5+n6+n7+n8+n9+p0+p1+p2+p3+p4
XM2I1 3 cout G2 Fets=n10+p11
XM2I2 9 sum G2 Fets=n27+p19
.ENDS
```

Из полученного SPICE-описания видно, что входящие в схему сумматора вентили XM0I1 (типа G0) и XM1I1 (типа G1) имеют закороченные входы. У семивходового вентиля G0 в схеме закорочены три пары входов: A с D, B с E и C с F. У пяти-входового вентиля G1 в схеме закорочены две пары входов: A с D и B с E.

### *B. Извлечение блока логической сети из транзисторной схемы*

Найденные вентили составляют единственную связную логическую сеть, у которой можно определить входы и выходы. Логическая сеть, извлеченная из двухуровневого SPICE-описания транзисторной схемы, приведена на рис. 3.

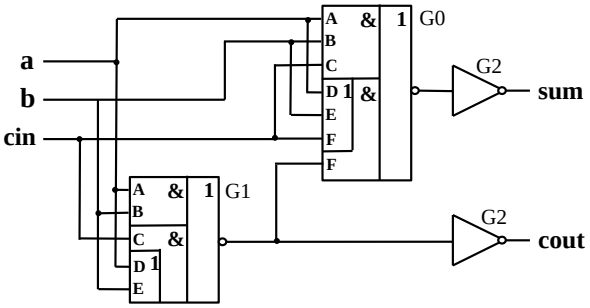


**Рис. 3. Логическая сеть одноразрядного сумматора**

Формат SPICE не содержит средств указания функций выводов сети. Чтобы отобразить эту информацию в SPICE-описании, логическая сеть выделяется как отдельная модель (схема C0), ее параметры, имена которых начинаются с символа «P», задают входы, с символа «O» – выходы схемы. Схема C0 имеет 5 выводов, три из которых являются входами. Соответствие параметров схемы C0 параметрам исходного двухуровневого SPICE-описания транзисторной схемы adder задается экземпляром XC0 (компонента типа C0) в сгенерированной схеме adder_gen (листинг 6).

**Листинг 6. SPICE-описание логической сети**

```
.SUBCKT C0 P0 P1 P2 O3 O4
XM0I1 P0 P1 P2 P0 P1 P2 1 2 G0
XM1I1 P0 P1 P2 P0 P1 1 G1
XM2I1 1 O3 G2
XM2I2 2 O4 G2
.ENDS
.SUBCKT adder_gen a b cin cout sum
XC0 a b cin cout sum C0
.ENDS
```

### *C. Нахождение функционального описания логической сети и верификация описаний*

После получения SPICE-описания логической сети производится его перевод на язык SF иерархических структурно-функциональных описаний дискретных устройств [5]. Поскольку выделенная схема является комбинационной, то ее функциональное описание представлено в формате LOG (логических уравнений) языка SF (листинг 7), которое переведено на язык VHDL (листинг 8).

**Листинг 7. Описание логической сети в формате LOG**

```
TITLE C0
FORMAT SF
AUTHOR extractor
DATE 25-05-2021
PROJECT adder_gen
DCL_PIN
EXT
INP
P0  P1  P2
OUT
O3  O4
INTER
END_PIN
FUNCTION
LOG
3 2 1
O3 = ^s1;
```

```
    O4 = ^(^((P0 * P1 * P2) + (s1 * (P0 + P1 +
P2))));
    s1 = ^((P0 * P1) + (P2 * (P0 + P1)));
    END_LOG
    END_FUNCTION
    END_C0
```

**Листинг 8. Описание логической сети на VHDL**

```
library IEEE;
use ieee.Std_Logic_1164.all;

entity C0 is
  port (
    O3 : out std_logic;
    O4 : out std_logic;
    P0 : in std_logic;
    P1 : in std_logic;
    P2 : in std_logic
  );
end entity C0;

architecture sf of C0 is
signal s1: std_logic;

begin
O3<=NOT s1;
O4<=NOT (NOT ((P0 AND P1 AND P2) OR (s1 AND (P0 OR P1
OR P2))));
s1<=NOT ((P0 AND P1) OR (P2 AND (P0 OR P1)));

end architecture sf;
```

Найденная логическая сеть реализует уравнения:

$$O_3 = P_0P_1 + P_2(P_0 + P_1),$$

$$O_4 = P_0P_1P_2 + \overline{O_3}(P_0 + P_1 + P_2).$$

После получения логической сети произведена ее верификация относительно исходного плоского SPICE-описания транзисторной схемы на топологическом уровне и на функциональном уровне относительно функционального описания из обзора [11]:

$$C_{k+1} = A_kB_k + C_k(A_k + B_k),$$

$$Sum_{k+1} = (A_k + B_k + C_k)\overline{C_{k+1}} + A_kB_kC_k .$$

Приведенные пары уравнений совпадают с точностью до обозначений и порядка термов.

## VI. Пример 2 декомпиляции транзисторной схемы

В качестве второго примера рассмотрим декомпиляцию транзисторной схемы полного одноразрядного сумматора, приведенной в разделе 11.2.1 монографии [12] (рис. 4). В этой схеме сеть из *p*-МОП-транзисторов идентична сети из *n*-МОП-транзисторов, т.е. не реализует инверсию проводимости *n*-МОП блока. Такая топология называется зеркальным сумматором. Эта топология уменьшает количество последовательно соединенных транзисторов и делает схему более простой и единообразной. Возможность такой организации топологии обусловлена тем, что функция сложения симметрична. Зеркальный сумматор (также как и схема сумматора из [11]) имеет большую задержку для вычисления суммы *Sum*, чем для вычисления переноса *Cout*. Spice-описание транзисторной схемы зеркального сумматора приведено на листинге 9.

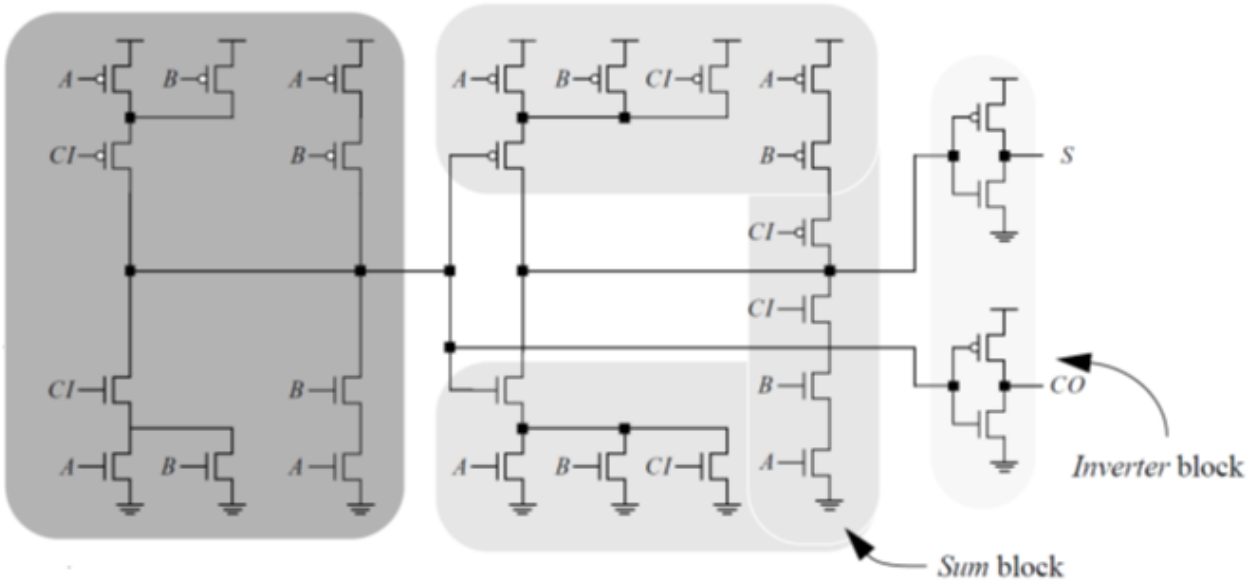


**Рис. 4. Транзисторная схема полного одноразрядного сумматора**

**Листинг 9. SPICE-описание зеркального сумматора**

```
.SUBCKT adder_book A B CI CO S vdd gnd
M0 vdd A 1 vdd p          M14 vdd A 7 vdd p M15
M1 1 CI 2 vdd p           7 B 8 vdd p
M2 vdd B 1 vdd p          M16 8 CI 9 vdd p
M3 vdd A 5 vdd p          M17 9 CI 10 gnd n
M4 5 B 2 vdd p            M18 10 B 11 gnd n
M5 2 CI 3 gnd n           M19 11 A gnd gnd n
M6 3 A gnd gnd n          M20 12 CI gnd gnd n
M7 3 B gnd gnd n          M21 12 B gnd gnd n
M8 2 B 4 gnd n            M22 12 A gnd gnd n
M9 4 A gnd gnd n          M23 9 2 12 gnd n
M10 vdd A 6 vdd p         M24 vdd 9 S vdd p
M11 6 2 9 vdd p           M25 S 9 gnd gnd n
M12 vdd B 6 vdd p         M26 vdd 2 CO vdd p
M13 vdd CI 6 vdd p        M27 CO 2 gnd gnd n
                          .ENDS
```

В результате декомпиляции транзисторной схемы обнаружено четыре группы транзисторов, представ-

ляющих схемы логических вентилей (листинг 10). Два вентиля из четырех, обнаруженных в результате декомпиляции, являются инверторами, два остальных реализуют функции:

$$\overline{A\,(B \vee C) \vee D\,E} \text{ и } \overline{ABC \vee G(D \vee E \vee E)}.$$

**Листинг 10. Результат декомпиляции**

```
Contents of circuit adder_book.sp:    Circuit: 'adder_book '
   Circuit adder_book contains 28 device instances.
     Class: n            instances:  14
     Class: p            instances:  14
   Circuit contains 19 nets.
   Connected Componens = 4
   Valid Componens = 4
   Pass fets = 0
   Psevdo Componens = 0 nets =0
   Unclassified fets = 0 nets = 0
   ((A AND (B OR C)) OR (D AND E))              1
   ((A AND B AND C) OR (G AND (D OR E OR F)))   1
   A         2
```

Результатом декомпиляции является двухуровневая транзисторная схема, представляемая иерархическим SPICE-описанием (листинг 11).

**Листинг 11. SPICE-описание двухуровневой транзисторной схемы зеркального сумматора**

```
* SPICE deck for cell adder_book_gen
.GLOBAL  vdd  gnd
.SUBCKT G0 A B C D E Y
* ((A AND (B OR C)) OR (D AND E))
M1 Y A 2 gnd n
M2 2 B gnd gnd n
M3 2 C gnd gnd n
M4 Y D 4 gnd n
M5 4 E gnd gnd n
M6 vdd B 6 vdd p
M7 6 A Y vdd p
M8 vdd C 6 vdd p
M9 vdd E 7 vdd p
M10 7 D Y vdd p
.ENDS
.SUBCKT G1 A B C D E F G Y
* ((A AND B AND C) OR (G AND (D OR E OR F)))
M1 Y A 2 gnd n
M2 2 B 3 gnd n
M3 3 C gnd gnd n
M4 5 D gnd gnd n
M5 5 E gnd gnd n
M6 5 F gnd gnd n
M7 Y G 5 gnd n
M8 vdd C 7 vdd p
M9 7 G Y vdd p
M10 vdd B 7 vdd p
M11 vdd A 7 vdd p
M12 vdd F 8 vdd p
M13 8 E 9 vdd p
M14 9 A Y vdd p
.ENDS
.SUBCKT G2 A Y
* A
M1 Y A gnd gnd n
M2 vdd A Y vdd p
.ENDS
.SUBCKT C0 P0 P1 P2 O3 O4
XM0I1 P2 P0 P1 P1 P0 1 G0
XM1I1 P2 P1 P0 P2 P1 P0 1 2 G1
XM2I1 1 O3 G2
XM2I2 2 O4 G2
.ENDS
.SUBCKT adder_book_gen A B CI CO S
XC0 A B CI CO S C0
.ENDS
```

Очевидно, что транзисторные схемы adder.sp и adder_book.sp топологически не эквивалентны, не эквивалентными оказываются и скомпилированные двухуровневые схемы adder_gen.sp и adder_book_gen.sp. Однако функционально пары этих схем эквивалентны.

## VII. Заключение

Разработанная программа декомпиляции протестирована на практических схемах транзисторного уровня и имеет достаточное быстродействие, чтобы обрабатывать схемы более чем с 100 тысячами транзисторов за несколько минут на персональной ЭВМ. Самая большая из исследованных схем содержала 345301 транзистор и декомпилировалась за 163 секунды на компьютере с чётырехъядерным процессором Intel i5-4460 3.20GHz и оперативной памятью 16,0 ГБ ЭВМ.

## Литература


[1] Rostami M., Koushanfar F., and Karri R. A primer on hardware security: Models, methods, and metrics // Proc. IEEE, vol. 102, no. 8, Aug. 2014. P. 1283–1295.

[2] Botero U.J. Hardware trust and assurance through reverse engineering: A survey and outlook from image analysis and machine learning perspectives / Botero U.J., Wilson R., Lu H., Rahman M.T., Mallaiyan M.A., Ganji F. .// arXiv preprint arXiv:2002.04210.

[3] Белоус А.И., Солодуха В. А. Основы кибербезопасности. Стандарты, концепции, методы и средства обеспечения. – Москва: ТЕХНОСФЕРА, 2021. 482 с.

[4] Tehranipoor M., Koushanfar F. A survey of hardware trojan taxonomy and detection // IEEE design & test of computers. 2010. V. 27. № 1. P. 10–25.

[5] Бибило П.Н., Романов В.И. Логическое проектирование дискретных устройств с использованием продукционно-фреймовой модели представления знаний. – Минск: Беларус. навука, 2011. 279 с.

[6] Бибило П.Н., Авдеев Н.А., Кардаш С.Н., Кириенко Н.А., Ланкевич Ю.Ю., Логинова И.П., Романов В.И., Черемисинов Д.И., Черемисинова Л.Д. Система логического проектирования функциональных блоков заказных КМОП СБИС с пониженным энергопотреблением // Микроэлектроника. 2018. Т. 47. № 1. С. 71–87.

[7] Черемисинов Д.И., Черемисинова Л.Д. Декомпиляция КМОП схемы из транзисторов в формате SPICE // Проблемы разработки перспективных микро- и наноэлектронных систем. 2018. Выпуск 1. С. 2-8. doi:10.31114/2078-7707-2018-1-2-8

[8] Черемисинов Д.И., Черемисинова Л.Д. Распознавание логических вентилей в плоской транзисторной схеме // Информатика. 2021. Т. 18. № 4. С. 54–65. https://doi.org/10.37661/1816-0301-2021-18-4-54-65.

[9] Черемисинов Д.И., Черемисинова Л.Д. Поиск часто встречающихся подграфов в графе, задающем схему на транзисторном уровне // Танаевские чтения: доклады Девятой Междун. научн. конф. (29.03–30.03 2021 г., Минск). – Мн.: ОИПИ НАН Беларуси. 2021. С. 132–136.

[10] Baker R.J. CMOS Circuit Design, Layout, and Simulation, Third Edition – Wiley-IEEE Press, 2010. – 1214 p.

[11] CMOS Binary Full Adder. A Survey of Possible Implementations // URL: http://web.engr.uky.edu/~elias/projects/10.pdf (дата обращения: 25.04.2022).

[12] Weste N. H. E., Harris D. M. CMOS VLSI Design: A Circuits and Systems Perspective. – Boston: Pearson/Addison-Wesley, 2010. – 867 p.

# Reverse Engineering of VLSI for Equipment Safety

D.I. Cheremisinov, L.D. Cheremisinova

The United Institute of Informatics Problems of the NAS of Belarus, Minsk

{cher, cld}@newman.bas-net.by

***Abstract* — VLSI reverse engineering is a powerful tool used in design verification to increase the performance of hardware simulation tools, as well as in detection of injected hardware illegal insets in the design and manufacturing process. The task of VLSI reverse engineering is to re-create the device specification by analyzing its hardware implementation in the form of VLSI. Reverse engineering is based on the decompilation of a flat transistor circuit netlist, which consists in recognition (extraction) of high-level structures in circuits at the transistor level and obtaining a representation at the logical level, equivalent to the original flat description at the transistor level. The most common style of logic is considered – logical complementary MOS structures.**

**Graph based methods and software tools are proposed for solving some key problems arising at the decompilation stage of transistor circuits: partitioning a graph into connectivity components corresponding to transistor subcircuits; recognition of subcircuits that are logical elements, and functions implemented by them; recognizing topologically equivalent transistor subcircuits; forming a library of recognized gates and constructing first two-level transistor circuit, then functionally equivalent logical network. It is assumed that the decompiled circuit, in addition to recognizable CMOS elements and pass gates, may also contain other structures that are distinguished during decompilation as pseudo-elements (not CMOS gates), as well as individual transistors. The original flat and resulting two-level transistor circuits are presented in SPICE format, logical networks in SF, VHDL or Verilog formats.**

**The developed decompilation program has been tested on practical transistor-level circuits and has sufficient speed to process circuits with more than 100 thousand transistors in a few minutes on a personal computer. Two practical examples of reverse engineering of transistor circuits are described.**

***Keywords* — CMOS transistor circuits, transistor subcircuit extraction, logic gate recognition, SPICE format.**

## REFERENCES

[1] Rostami M., Koushanfar F., and Karri R. A primer on hardware security: Models, methods, and metrics // Proc. IEEE, vol. 102, no. 8, Aug. 2014. P. 1283–1295.

[2] Botero U.J. Hardware trust and assurance through reverse engineering: A survey and outlook from image analysis and machine learning perspectives / Botero U.J., Wilson R., Lu H., Rahman M.T., Mallaiyan M.A., Ganji F. .// arXiv preprint arXiv:2002.04210.

[3] Belous A.I., Solodukha V.A. Osnovy kiberbezopasnosti. Standarty, kontseptsii, metody i sredstva obespecheniya (Fundamentals of Cybersecurity. Standards, Concepts, Methods and Means of Support) – Moscow: Tekhnosfera, 2021. 482 p. (In Russ.).

[4] Tehranipoor M., Koushanfar F. A survey of hardware trojan taxonomy and detection // IEEE design & test of computers. 2010. V. 27. № 1. P. 10–25.

[5] Bibilo P.N., Romanov V.I. Logicheskoye proyektirovaniye diskretnykh ustroystv s ispol'zovaniyem produktsionno-freymovoy modeli predstavleniya znaniy (Logical design of discrete devices using a production-frame model of knowledge representation). – Minsk: Belarus. navuka, 2011. 279 p.

[6] Bibilo P.N., Avdeyev N.A., Kardash S.N., Kiriyenko N.A., Lankevich YU.YU., Loginova I.P., Romanov V.I., Cheremisinov D.I., Cheremisinova L.D. Sistema logicheskogo proyektirovaniya funktsional'nykh blokov zakaznykh KMOP SBIS s ponizhennym energopotrebleniyem (System of logical design of functional blocks of custom CMOS VLSI with reduced power consumption) // Mikroelektronika. 2018. Vol. 47. № 1. P. 71–87.

[7] Cheremisinov D.I., Cheremisinova L.D. Decompilation of Flat CMOS Circuits in SPICE Format // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2018. Issue 1. P. 2-8. doi:10.31114/2078-7707-2018-1-2-8(In Russ.)

[8] Cheremisinov D.I., Cheremisinova L.D. Raspoznavaniye logicheskikh ventiley v ploskoy tranzistornoy skheme (Logical gates recognition in a flat transistor circuit) // Informatika [Informatics]. 2021. Vol. 18. № 4. P. 83−94. DOI: 10.37661/1816-0301-2021-18-4-54-65 (in Russian).

[9] Cheremisinov D.I., Cheremisinova L.D. Poisk chasto vstrechayushchikhsya podgrafov v grafe, zadayushchem skhemu na tranzistornom urovne (Search for frequently occurring subgraphs in a graph defining a circuit at the transistor level) // Tanayevskiye chteniya: doklady Devyatoy Mezhdun. nauchn. konf. (29.03–30.03 2021 g., Minsk). – Mn.: OIPI NAN Belarusi. 2021. P. 132–136.

[10] Baker R.J. CMOS Circuit Design, Layout, and Simulation, Third Edition – Wiley-IEEE Press, 2010. – 1214 p.

[11] CMOS Binary Full Adder. A Survey of Possible Implementations // URL: http://web.engr.uky.edu/~elias/projects/10.pdf (дата обращения: 25.04.2022).

[12] Weste N. H. E., Harris D. M. CMOS VLSI Design. A Circuits and Systems Perspective. – Boston: Pearson/Addison-Wesley, 2010. – 867 p.