

УДК 621.3.049.77.001.2

С. Э. Миронов, А. О. Монько

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Средства автоматизации плотноупакованного проектирования топологии ячеек СБИС

Описаны средства проектирования плотноупакованной топологии ячеек интегральных схем. Разработанные программные средства генерируют технологически инвариантное описание эскизов топологии, которое с помощью средств плотноупакованного сжатия топологии настраивается на требуемые пользователю проектные нормы.

Технологически инвариантное проектирование топологии, сжатие топологии, изменение формы элементов, автоматизация

Технологически инвариантное плотноупакованное проектирование топологии СБИС. Увеличение сложности микронных проектов и ужесточение предъявляемых к ним требований по быстродействию делает необходимым постоянное совершенствование технологии изготовления СБИС, выражающееся в уменьшении проектных норм. Улучшение же характеристик интегральных схем посредством совершенствования технологических процессов приводит к необходимости полной переработки топологии библиотек ячеек и блоков СБИС.

В связи с этим одной из основных отличительных особенностей современных микронных проектов стала технологическая инвариантность – возможность настройки проекта на любые проектные нормы. Она достигается с помощью систем сжатия топологии, минимизирующих расстояние между элементами в соответ-

ствии с проектными нормами и задаваемыми разработчиком ограничениями на расположение отдельных частей топологии.

Наибольшая плотность упаковки топологии интегральных схем (одно из основных требований, предъявляемых к микронным проектам) достигается, когда затворы транзисторов (рис. 1, а) изменяют свою форму и «обвивают» контакты (рис. 1, б), и на затворах образуются колена.

Ранее минимизация площади схем посредством изменения формы транзисторов считалась прерогативой человека [1], и решение данной проблемы в САПР БИС считалось невозможным. Однако оказалось, что возможность придания транзисторам в процессе сжатия сложной формы (рис. 2, а) достигается их представлением в виде сборки из секций затворов, секций областей легирования и контуров активных областей транзи-

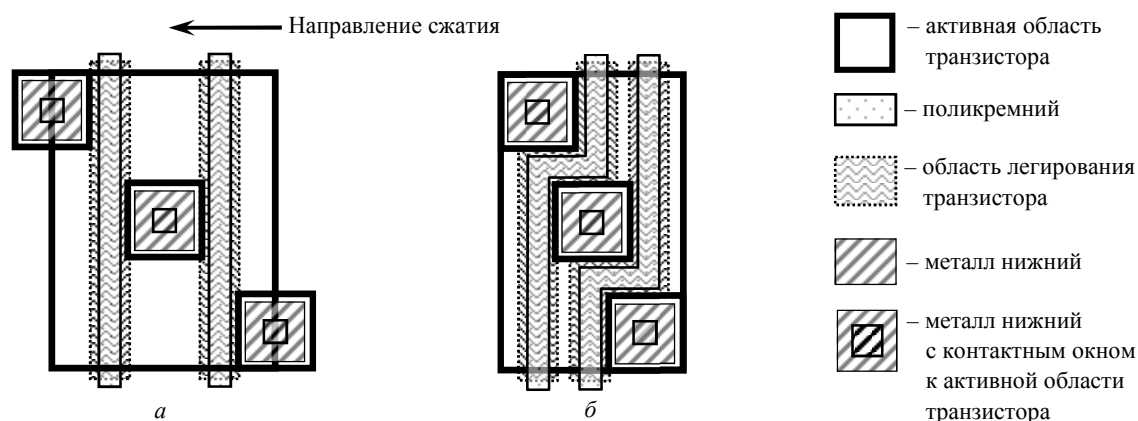


Рис. 1

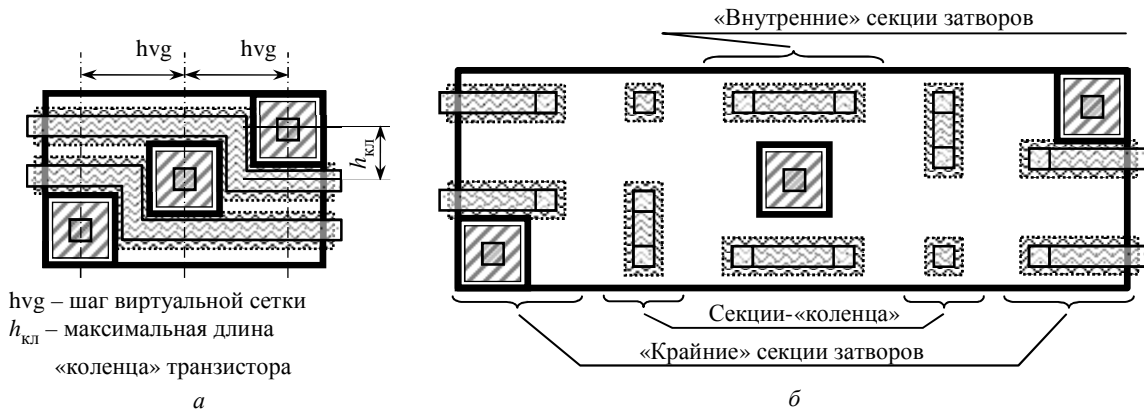


Рис. 2

сторов, но не индивидуальных для каждого транзистора или тем более секции, а общих для всей сборки (рис. 2, б) [2]. При этом упрощается процесс сжатия, как за счет упрощения алгоритма, так и за счет уменьшения числа объектов сжатия.

Автоматизация плотноупакованного проектирования топологии ячеек СБИС. В связи с постоянным ростом темпов развития и сложности вычислительных устройств и систем в микроэлектронике огромное значение приобрел фактор обеспеченности средствами САПР. Применительно к проектированию топологии вообще и к системам сжатия топологии в частности автоматизация в первую очередь ассоциируется с редактором топологии.

Экспериментальная версия редактора топологии, описываемая в данной статье, служит для ввода эскиза топологии как графически, с помощью ин-

струментов редактора, так и посредством импорта из текстового файла. Редактор имеет возможности масштабирования и редактирования элементов.

На рис. 3 представлены окна экспериментальной версии редактора топологии с эскизами транзисторной сборки: исходной (рис. 3, а) и сжатой с изменением формы затворов (рис. 3, б).

Зачастую удобнее добавлять элементы интерактивно с помощью операций растягивания и перемещения по сетке мышью и клавиатурой. Однако помимо ввода эскиза топологии графически (с помощью соответствующих инструментов редактора) для точного позиционирования элементов и прецизионного редактирования их параметров предусмотрено добавление и редактирование элементов и через диалоги ввода (рис. 4), позволяющее пользователю в том числе и просто уточнить параметры элемента.

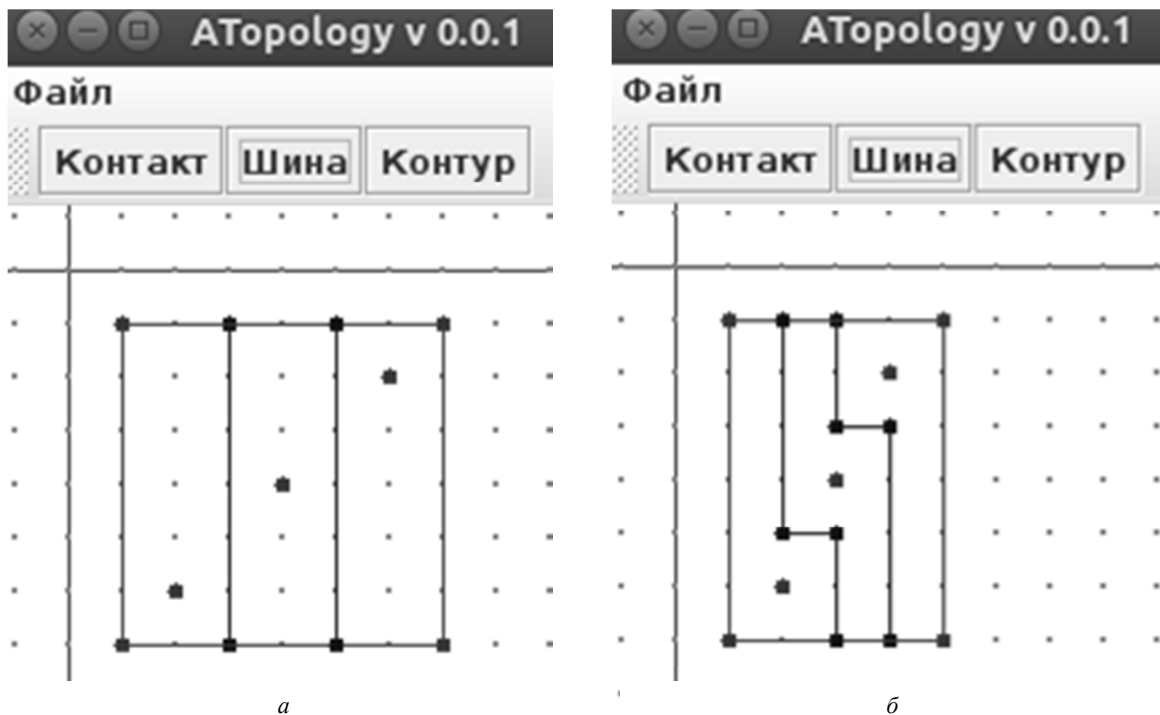


Рис. 3

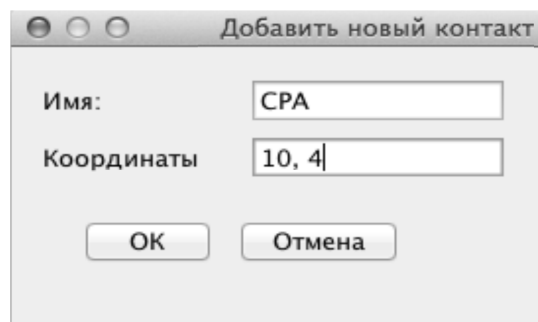


Рис. 4

Форматы данных системы плотноупакованного проектирования топологии ячеек. Для исходной и результирующей топологий в проектируемой системе сжатия с изменением формы элементов планируется набор уровней хранения данных как в оперативной памяти (во время обработки), так и в файлах. Эти уровни предполагается сделать полностью взаимосвязанными и двухсторонними, однако центром этой подсистемы являются внутренние структуры представления данных топологии.

Для описания исходных технологически инвариантных эскизов топологии в системе предусмотрен упрощенный текстовый формат для быстрого и компактного представления данных в виртуальных координатах. Топология в нем задана набором элементов с минимально необходимой информацией для их представления и дальнейшей обработки (в том числе корректировки и уточнения) в графическом редакторе. К такой информации относятся:

- тип элемента, задаваемый его именем;
- координаты точки привязки контактов, координаты угловых точек контуров или осевых линий шин;
- ширина шин, задаваемая в минимально допустимых проектными нормами ширинах.

В приведенных ниже примерах

CPA(4, 5); // Контакт PA-типа

ZNSI W(2) (1, 2; 1, 4); // Затвор транзистора N-типа;

- KP и ZNSI – имена элементов;
- разделенные запятыми пары чисел – координаты точек точки привязки контакта между активной областью p -типа и нижним металлом и координаты осевой линии затвора транзистора n -типа;
- заключенное в скобки число после символа W – ширина шины затвора.

Более детальным форматом описания топологии является формат представления структур в текстовых файлах (xml, json). Это формат, который предполагает явное указание связей между элементами как посредством иерархической вложенности, так и посредством ссылок. Это дости-

гается за счет того, что каждый элемент имеет уникальный идентификатор – имя, которое может быть задано пользователем или по умолчанию сгенерировано системой. Такой формат предполагает некоторое знание внутренней структуры данных и более детальное конфигурирование отдельных частей топологии.

В приведенном ниже примере описывается топология шины, состоящей из одного отрезка.

```
<topology class="VirtualGrid" step="20.0">
  <Wire name="sd">
    <parts>
      <part name="1760857141">
        <axis>
          <start x="0.0" y="0.0" />
          <end x="0.0" y="5.0" />
        </axis>
        <index>0</index>
        <width>0.0</width>
        <widthAtBorder>0.0</widthAtBorder>
        <maxLength>1.7976931348623157E308</maxLength>
      </part>
    </parts>
  </Wire>
</topology>
```

Большинство из полей в таком описании может отсутствовать, обязательным для каждого элемента должно быть имя. С учетом этого можно сократить пример до более компактного вида.

```
<topology class="VirtualGrid" step="20.0">
  <Wire name="sd">
    <parts>
      <part name="1760857141">
        <axis>
          <start x="0.0" y="0.0" />
          <end x="0.0" y="5.0" />
        </axis>
      </part>
    </parts>
  </Wire>
</topology>
```

Несмотря на то, что такой формат в основном предназначен для внутренних нужд системы, предполагается импорт топологии, представленной в описанном формате. Помимо xml-вида предполагается иметь возможность работать с json-документами (см. ниже).

```
Topology :
{
Class : "VirtualGrid";
Wire :
{
Name : "sd";
Parts :
{
...
}
}
}
```

Для быстрого сохранения текущего состояния топологии в процессе сжатия и сохранения его

ние промежуточных результатов сжатия отдельных частей топологии на различных этапах, а не иметь дело с более долгим процессом чтения/записи текстовых форматов.

Результаты плотноупакованного сжатия топологии ячеек планируется сохранять и в стандартных форматах текстовых и бинарных языков описания топологии (CIF, GDSII и др.).

На рис. 5 представлена структура планируемой подсистемы ввода/вывода топологии в виде диаграммы взаимодействия форматов описания топологии.

Видно, что все данные проходят промежуточный этап преобразования в xml/json-структуры. Это сделано для того, чтобы можно было легко добавлять новые форматы и преобразование не было жестко привязано к внутренней структуре классов, а также для тестирования и отладки системы.

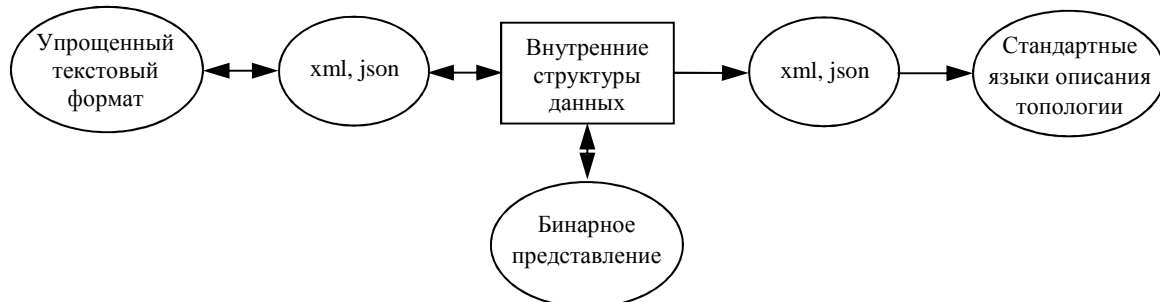


Рис. 5

для последующей обработки предназначен бинарный формат. В частности, предполагается использовать механизм java serialization, в котором внутренние структуры данных представляются массивом байт в зависимости от состояния полей. Такой подход обеспечит плотноупакованное проектирование, так как позволит реализовать необходимое для обработки возникающих при сжатии особых ситуаций [3] быстрое сохранение и чтение

Наряду с вводом и отображением самой топологии редактор предполагает и ввод проектных норм посредством их импорта из текстового файла, содержащего для конкретной технологии данные о минимально возможных значениях ширины элементов в разных топологических слоях и зазорах между ними.

СПИСОК ЛИТЕРАТУРЫ

1. Croes, K. CAMELEON: A Process Tolerant Symbolic Layout / K. Croes, H. J. De Man, P. Six // IEEE J. of Solid-State Circuits. 1988. Vol. 23, № 3. P. 705–713.
2. Миронов С. Э., Монько А. О. Сжатие топологии с изменением формы элементов // Изв. СПбГЭТУ «ЛЭТИ». 2013. № 4. С. 29–35.

3. Миронов С. Э., Монько А. О., Сафьянников Н. М. Особые ситуации при сжатии топологии фрагментов СБИС с изменением формы транзисторов // Изв. СПбГЭТУ «ЛЭТИ». 2013. № 5. С. 79–84.

S. E. Mironov, A. O. Mon'ko
Saint-Petersburg state electrotechnical university «LETI»

AUTOMATION TOOL FOR HIGH DENSITY LAYOUT DESIGN OF VLSI CELLS

The article is devoted to the description the design tools high density layout of integrated circuits cell. Developed software tools generate description of process tolerant layout sketches, which are configured on the user required design rules by high density layout compaction tool.

Process tolerant design of layout, layout compaction, element form change, automation