

**АО «НПЦ СпецЭлектронСистемы»**

**Программное обеспечение «Система поддержки МКП для испытаний и сертификации» (Программные средства для интеграции в топологию разрабатываемой СБИС средств проверки соответствия переданной на фабрику информации и полученного изделия)**

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**ЛИСТОВ 20**

## **АННОТАЦИЯ**

В настоящем документе содержатся сведения о программном обеспечении «Система поддержки МКП для испытаний и сертификации» (Программные средства для интеграции в топологию разрабатываемой СБИС средств проверки соответствия переданной на фабрику информации и полученного изделия) для пользователя программы в части работы с программой, в следующем объеме:

- условия выполнения программы;
- выполнение программы.

Настоящий документ предназначен для инженерно-технических специалистов, осуществляющих эксплуатацию, поддержку и сопровождение программного изделия.

## Содержание

<b>1 Введение</b>	<b>4</b>
<b>2 Термины и определения</b>	<b>5</b>
<b>3 Общие положения</b>	<b>7</b>
3.1 Наименование программы	7
3.2 Область применения программы	7
3.3 Назначение программы	7
<b>4 Условия выполнения программы</b>	<b>8</b>
4.1 Режимы работы программы	8
4.2 Требования к уровню подготовки пользователя	8
4.3 Требования к ПК из состава АРМ пользователя	8
4.4 Программное обеспечение, необходимое для функционирования программы	9
<b>5 Выполнение программы</b>	<b>10</b>
5.1 Подключение к программе. Вход в программу	10
5.1.1 Организация рабочих директорий	10
5.2 Описание алгоритма	10
5.3 Входные и выходные данные	19
5.3.1 Подготовка входных данных: GDS	19
5.3.2 Подготовка входных данных: послойные фотографии кристалла	20

# 1 Введение

В настоящем документе содержатся сведения о программном обеспечении «Система поддержки МКП для испытаний и сертификации» (Программные средства для интеграции в топологию разрабатываемой СБИС средств проверки соответствия переданной на фабрику информации и полученного изделия) для пользователя программы в части работы с программой, в следующем объёме:

- условия выполнения программы;
- выполнение программы.

Настоящий документ предназначен для инженерно-технических специалистов, осуществляющих эксплуатацию, поддержку и сопровождение программного изделия.

## 2 Термины и определения

В настоящем документе используются термины и основные понятия области автоматизированных систем, определённые в ГОСТ Р 59853-2021 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения», а также введённые термины и определения, представленные в таблице Таблица 1.

Таблица 1 – Термины и определения

Термин	Сокращение	Определение
Автоматизированное рабочее место	АРМ	Программно-технический комплекс, предназначенный для автоматизации деятельности определенного вида. Объединяет программно-аппаратные средства, обеспечивающие взаимодействие человека с компьютером
Авторизация	-	Процесс принятия решения о предоставлении доступа пользователю на выполнение операции на основании каких-либо знаний о пользователе
Аутентификация	-	Процесс проверки информации о пользователе
Общесистемное программное обеспечение	ОПО	Часть программного обеспечения Платформы, являющееся средой для функционирования прикладного программного обеспечения (ППО) и специального программного обеспечения (СПО) Платформы. К общесистемному программному обеспечению относятся операционные системы и программные средства виртуализации
Прикладное программное обеспечение	ППО	Часть программного обеспечения Платформы, не являющееся общесистемным программным обеспечением (ОПО), но необходимое для функционирования специального программного обеспечения (СПО) Платформы
Программный модуль	-	Фрагмент описания процесса, оформляемый как отдельная часть программного кода, который полностью самостоятельно выполняет свою задачу, оформленный в виде отдельного файла с исходным кодом или поименованной непрерывной её части
Сервер	-	В контексте сетевой архитектуры «клиент-сервер»: программное обеспечение, являющееся поставщиком услуг заказчикам (клиентам)

Система управления базами данных	СУБД	Совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных
Специальное (прикладное) программное обеспечение	СПО	Часть программного обеспечения Платформы, разработанного в рамках создания Платформы, в том числе посредством настройки (адаптации) программного обеспечения, принадлежащего третьим лицам или Исполнителю
Учётная запись	-	Хранимая в компьютерной системе совокупность данных о пользователе, необходимая для его опознавания (аутентификации) и предоставления доступа к его личным данным и настройкам
Хост	-	Хост (англ. host) - любое устройство, предоставляющее сервисы формата «клиент-сервер» в режиме сервера по каким-либо интерфейсам и уникально определённое на этих интерфейсах. В более широком смысле под хостом могут понимать любой компьютер, подключённый к локальной или глобальной сети. Употребление слова «хост» имеет смысл только вместе с пояснением, хостом какого сервиса предполагается называемое устройство

## **3 Общие положения**

### **3.1 Наименование программы**

Полное наименование программного обеспечения: ««Система поддержки МКП для испытаний и сертификации» (Программные средства для интеграции в топологию разрабатываемой СБИС средств проверки соответствия переданной на фабрику информации и полученного изделия) (далее по тексту - «Программа»).

### **3.2 Область применения программы**

Областью применения наименование программного обеспечения ««Система поддержки МКП для испытаний и сертификации» (Программные средства для интеграции в топологию разрабатываемой СБИС средств проверки соответствия переданной на фабрику информации и полученного изделия) является деятельность в сфере проектирования, разработки и производства изделий электронной техники в следующих организациях и учреждениях:

- государственные учреждения, министерства и ведомства;
- негосударственные организации;
- коммерческие организации с различными видами собственности;
- некоммерческие организации.

### **3.3 Назначение программы**

Программное изделие предназначено для управления проектами разработки изделий электронной техники.

Программное изделие предназначено для установки и функционирования на аппаратных средствах, размещённых на указанных выше объектах, перечисленных в разделе 3.2 настоящего документа.

## 4 Условия выполнения программы

### 4.1 Режимы работы программы

Клиентское ПО представляет возможность работы с Программой в единственном режиме – в режиме пользователя. Данный режим доступен Оператору - Пользователю, которому назначена встроенная роль «Пользователь».

### 4.2 Требования к уровню подготовки пользователя

Для работы с Программой Пользователь должен обладать знаниями и навыками на уровне уверенного пользователя персональных компьютеров под управлением операционных систем Linux.

### 4.3 Требования к ПК из состава АРМ пользователя

Для доступа Пользователей к Программе предполагается использование автоматизированных рабочих мест (далее по тексту – «АРМ»).

В качестве АРМ может быть использовано следующее оборудование:

- стационарные персональные компьютеры в настольном исполнении;
- носимые персональные компьютеры-«ноутбуки».

Персональный компьютер (ПК) из состава АРМ пользователя должен иметь характеристики, представленные в Таблица 2.

Таблица 2 – Характеристики ПК из состава АРМ пользователя

Наименование	Характеристика
Тактовая частота процессор	Не менее 2 ГГц
Количество ядер процессора	Не менее двух
Оперативная память	Не менее 8 Гб
Свободное пространство на жестком диске	Не менее 50 Гб
Сетевой интерфейс проводной	Gigabit Ethernet
Клавиатура	Стандартная
Кнопочный манипулятор типа «Мышь»	Стандартный, с тремя кнопками
Монитор	Разрешение не менее 1920 x 1080

#### **4.4 Программное обеспечение, необходимое для функционирования программы**

Операционная система Linux. Допускается использование ОС любой версии.

Python: 3.4 и выше;

Модули Python: matplotlib, math, copy, re, fileinput;

Cadence® Innovus™ версии v18.14-s088\_1 в режиме stylus;

Cadence® Innovus™ на основе библиотеки стандартных ячеек tsmc28.

## 5 Выполнение программы

### 5.1 Организация рабочих директорий

Структура рабочей директории *.bisa\_py* представлена в Таблице 3.

Таблица 3 – Структура директории *bisa\_py*

Путь рабочей директории	Описание содержимого директории
<i>./bisa_py/tcl/</i>	tcl – скрипты с командами САПР Innovus
<i>./bisa_py/sh/</i>	sh – скрипты для работы со списками
<i>./bisa_py/src/</i>	Исходные данные для python-скрипта main.py
<i>./bisa_py/main.py</i>	Python-скрипт, выполняющий обработку списков
<i>./bisa_py/output</i>	Выходные данные после выполнения python-скрипта
<i>./bisa_py/BISA_REPORTS/ REPORTS_BEFORE_BISA</i>	Отчёты по временным параметрам hold/setup исходной топологии

### 5.2 Описание алгоритма

В процессе разработки топологии микросхемы оценивается плотность заполнения логическими ячейками площади функционального блока. Предполагается, что на начальном этапе плотность размещения элементов должна быть от 40% до 70% от всей площади. Оставшиеся пустые места заполняются ячейками, не имеющими логической функции.

Проконтролировать заполнение оставшейся площади предлагается заменой ячеек без логики на триггеры D-типа. Такое решение позволит оценить по входным сигналам ожидаемые результаты на выходе.

Алгоритм действий построен следующим образом<sup>1</sup>:

- в рабочую директорию проекта Cadence® Innovus™ *<install\_dir>* копируется директория ***bisa\_py***, содержащая скрипты, выполняющие автозаполнение топологии;
- запускается Cadence® Innovus™ через команду: *innovus -stylus* и считывается база данных проекта через команду: *read\_db [путь к базе данных]*;
- Запускается скрипт авторазмещения через команду: *source .bisa\_py/tcl/autorun.tcl*.

1 Рабочие директории и файлы отмечены жирным курсивом. Команды САПР Cadence® Innovus™ отмечены курсивом.

Далее выполняется последовательность скриптов<sup>2</sup>:

- На первом этапе, с помощью скрипта *.bisa\_py/tcl/get\_db.tcl*, собираются списки данных, содержащие информацию о всех ячейках проекта.

Данный скрипт предварительно удаляет ячейки типа *filler* из базы данных проекта с помощью команды *innovus delete\_filler*, затем происходит считывание базы данных командой *get\_db* по следующим параметрам:

*.name* – имя ячейки

*.location* – расположение ячейки по координатам

*.base\_cell.bbox.ur* – размер стандартной ячейки

*.place\_status* – статус размещения ячейки

*.rows.rect.ll* – координаты границ размещения

*.place\_blockages.rects* – координаты областей, где запрещено размещение

Полученный файл *.bisa\_py/src/inst\_attributes.list* содержит список с информацией следующего типа:

```
.name { .location } { .base_cell.bbox.ur } .place_status
```

Полученные файлы *.bisa\_py/src/rows.rect.\*.list* и *.bisa\_py/src/place\_blockages.rects.\*.list* содержат информацию о границах областей *core* и *place\_blockages*, необходимую для оптимальной работы алгоритма.

В конце данный tcl-скрипт вызывает shell-скрипт *.bisa\_py/sh/organize\_lists.sh* который производит разбиение полученных из Cadence® Innovus™ списков на списки удобного формата, с помощью команд *awk* и *sed*.

- На втором этапе происходит запись отчётов по временным параметрам *hold/setup* исходной топологии в директорию *.bisa\_py/BISA\_REPORTS/REPORTS\_BEFORE\_BISA* с помощью команд *innovus time\_design*.

- На третьем этапе выполняется python-скрипт *.bisa\_py/main.py*.

В начале работы данного скрипта оценивается свободное место в каждом ряду ячеек и опционально изменяются массивы их исходных координат для увеличения плотности размещения, используя координаты уже установленных ячеек и их перевод в удобный формат. Результатами работы данного python-скрипта являются новые списки с координатами внедряемых элементов *.bisa\_py/output/bisa.\*.list* и измененными координатами исходных ячеек *.bisa\_py/output/new.\*.list*, список соединений внедряемых элементов *.bisa\_py/output/bisa.nets.list*, а также файл с есо-директивами *.bisa\_py/output/eco\_file*, который позволяет дополнить netlist основного блока

---

2 Рабочие директории и файлы отмечены жирным курсивом. Команды САПР Cadence® Innovus™ отмечены курсивом.

логикой соединений между внедряемыми ячейками.

Ячейка D-триггера имеет фиксированную длину, поэтому для уменьшения количества пустых мест, предлагается сдвинуть координаты ячеек в Cadence® Innovus™, используя скрипт `./tcl/change_coord_stylus_ui.tcl`. Сдвиг ячеек может ухудшить временные характеристики, поэтому этот шаг является опциональным.

- Далее для внесения в базу данных Cadence® Innovus™ информации о новых соединениях запускается скрипт `./tcl/create_net.tcl`. Он использует команду `innovus create_net`, которая загружает полученный ранее список соединений `./bisa_py/output/bisa.nets.list`.

- Для загрузки информации о новых соединениях в netlist используется команда Cadence® Innovus™: `loadECO ./bisa_py/output/eco_file`, которая использует полученный ранее файл с есо-директивами `./bisa_py/output/eco_file`.

- Для размещения внедряемых ячеек в соответствии с координатами, полученными после выполнения скрипта `./bisa_py/main.py`, используется скрипт `./bisa_py/tcl/place_bisa_stylus_ui.tcl`.

- На финальном этапе проводится трассировка новых межсоединений с помощью команды Cadence® Innovus™ `ecoRoute` и оцениваются временные параметры новой схемы с записью отчётов в директорию `./bisa_py/BISA_REPORTS/REPORTS_AFTER_BISA`.

В результате выполнения авторазмещения должна появиться цепочка соединённых D-триггеров, которая включена в топологию вместо ячеек типа filler.

Схематично последовательность выполнения скриптов иллюстрирует Рисунок 1.

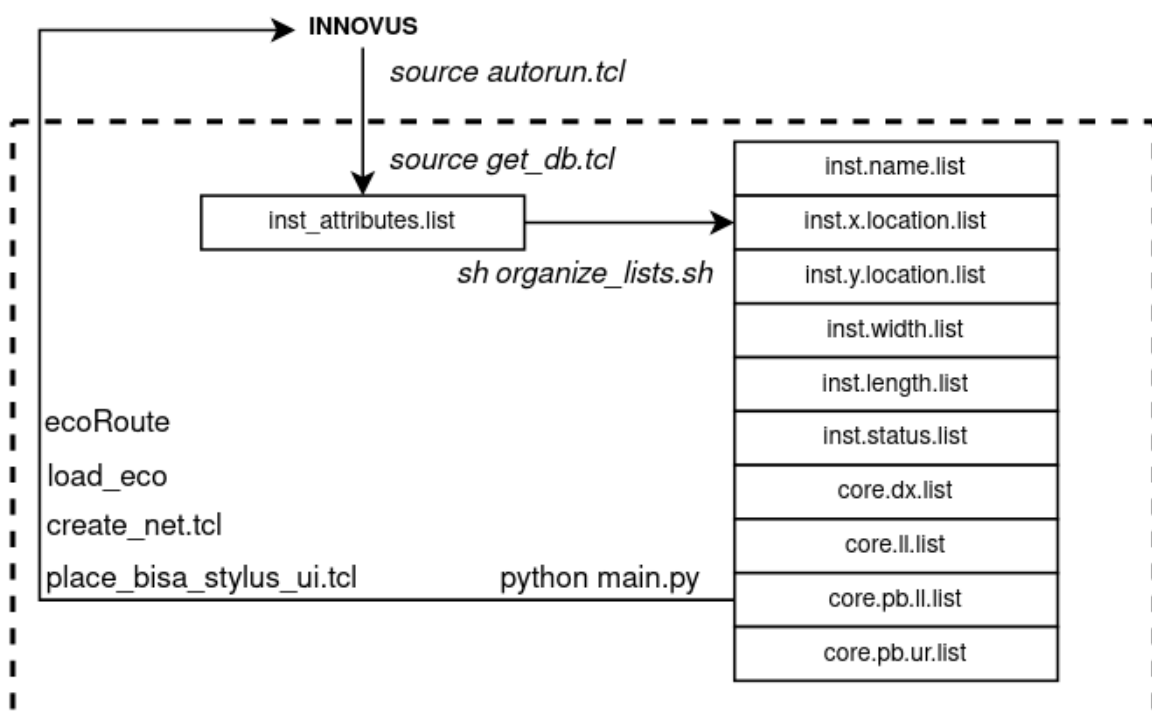


Рисунок 1 – Схема выполнения последовательности скриптов

## **6 Входные и выходные данные**

### **6.1 Входные данные**

Исходный проект в Cadence® Innovus™.

### **6.2 Выходные данные**

Измененная база проекта Cadence® Innovus™, содержащая средства проверки соответствия переданной на фабрику информации и полученного изделия.