

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет «Московский институт электронной техники».

Институт микроприборов и систем управления имени Л.Н. Преснухина

## **Лабораторная работа №8**

### **«Проектирование ключа на p-n-диодах»**

**Модуль**  
**«Моделирование приёмопередающих устройств»**

**По курсу**  
**«Моделирование СВЧ-устройств в САПР»**

Москва, Зеленоград

2024

## Оглавление

Оглавление .....	2
Введение.....	3
Теоретические сведения .....	4
Методика выполнения работы.....	7
Создание и первоначальная настройка проекта .....	7
Создание компонента-хранилища S-параметров PIN-диода .....	7
Выбор режимов работы pin-диода.....	16
Расчет токоограничивающего резистора .....	18
Проектирование SPST-ключа.....	19
Проектирование SPDT-ключа .....	24
Проектирование неотражающего SPDT-ключа .....	29
Задание на выполнение.....	33
Требования к отчёту.....	38
Задание на самостоятельную работу .....	39
Контрольные вопросы .....	40
Литература .....	41



## Введение

**Цель работы:** ознакомиться методикой проектирования СВЧ-ключей на pin-диодах в среде Keysight Advanced Design System (ADS).

**Используемое оборудование или ПО:** Keysight Advanced Design System 2023

**Продолжительность работы:** 4 часа.

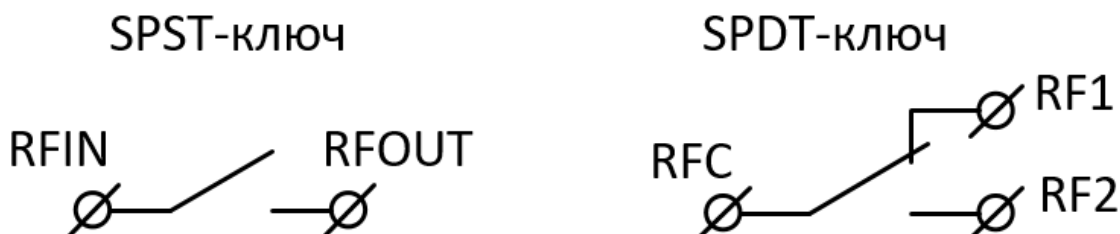
В разделе «Методика выполнения» приведены только необходимые действия по выполнению лабораторной работы. Подробно описываются только новые приемы работы в ADS. Предполагается, что студент выполнил предыдущие лабораторные работы.

<b>Lab1</b>	показывает места, которые подробно описаны в предыдущих лабораторных работах
	показывает новые приемы, значительно упрощающие или ускоряющие использование ADS
	показывает места, за которыми надо особенно следить и где легко совершить ошибку

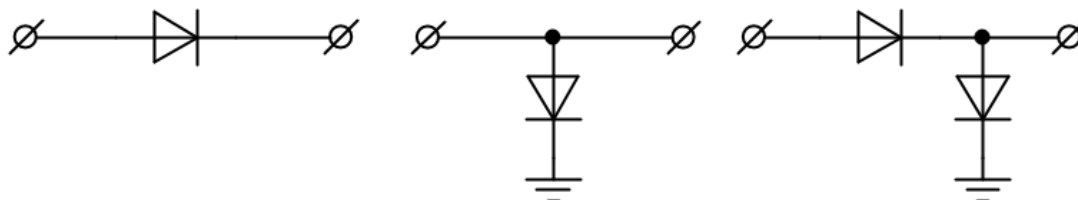
## Теоретические сведения

При проектировании СВЧ-ключевых устройств часто используются PIN-диоды. PIN-диоды имеют относительно широкий нелигированный регион, что за счет получаемой малой барьерной емкости эффективно их использовать как быстрый переключаемый элемент в ключах, управляемых дискретных аттенюаторах и фазовращателях и пр.

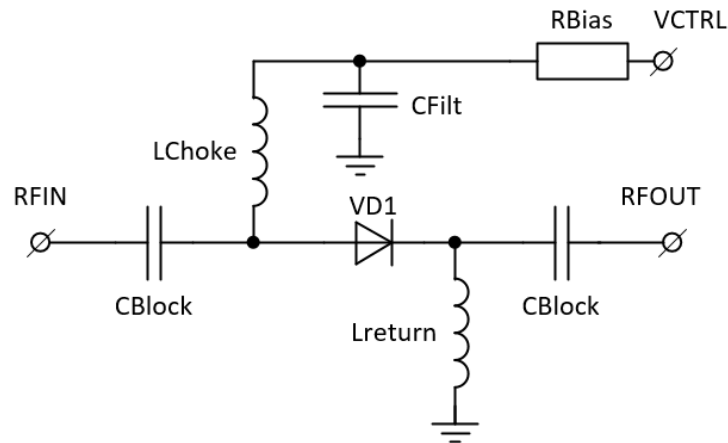
В международной литературе принято 4-х буквенное обозначение назначения ключа по его виду. Первые две буквы означают схему входов, вторые – выходов. Например, SPST расшифровывается как Single Pole Single Through, а SPDT – Single Pole Double Through.



В высокочастотном SPST-ключе PIN-диод может быть поставлен последовательно, параллельно или комбинировано. При последовательном включении открытый PIN-диод замыкает выводы между собой, закрытый – размыкает. При параллельном – наоборот, открытый PIN-диод замыкает линию с землей и создает в точке соединения короткое замыкание. Комбинированное последовательно-параллельное включение PIN-диодов используют, если необходимо усилить запирающее действие в закрытом режиме.



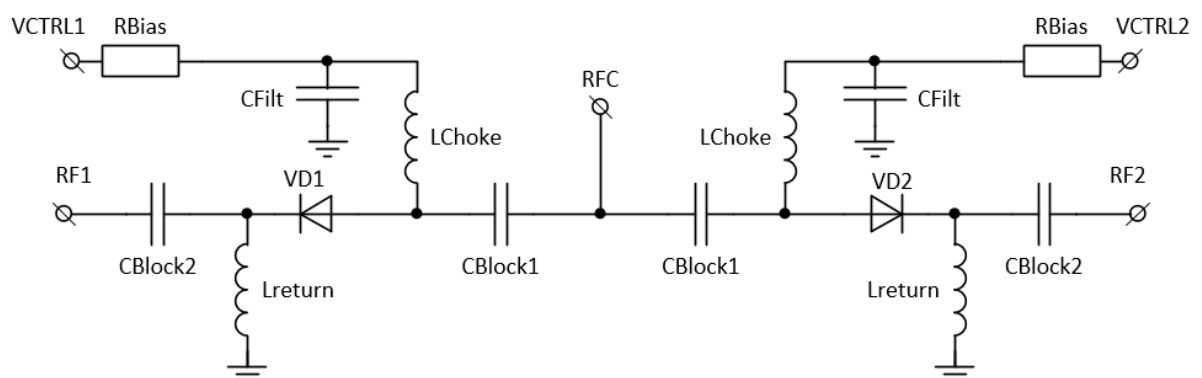
Одна из самых простых и базовых схем высокочастотного SPST-ключа с последовательным PIN-диодом показана ниже.



При подаче высокого управляющего напряжения  $VCTRL\_ON$ , постоянный ток течет через цепь  $RBias - Lchoke - VD1 - Lreturn$ . В этом режиме PIN-диод  $VD1$  открывается и имеет малые потери на проход. Резистор  $RBias$  выступает в качестве токоограничивающего (токозадающего) для задания прямого смещения  $VD1$ . Конденсаторы  $CBlock$  развязывают по постоянному току схему от соседей. Также LC-цепи  $CBlock/LChoke$  и  $Cblock/Lreturn$  используют для согласования PIN-диода.

При подаче 0 В или отрицательного управляющего напряжения PIN-диод  $VD1$  закрывается. В этом режиме он как правило имеет коэффициент отражения близкий к 0,9..1. Т.к. обратный ток PIN-диода мал, что влиянием резистора  $RBias$  на напряжение обратного смещения можно пренебречь.

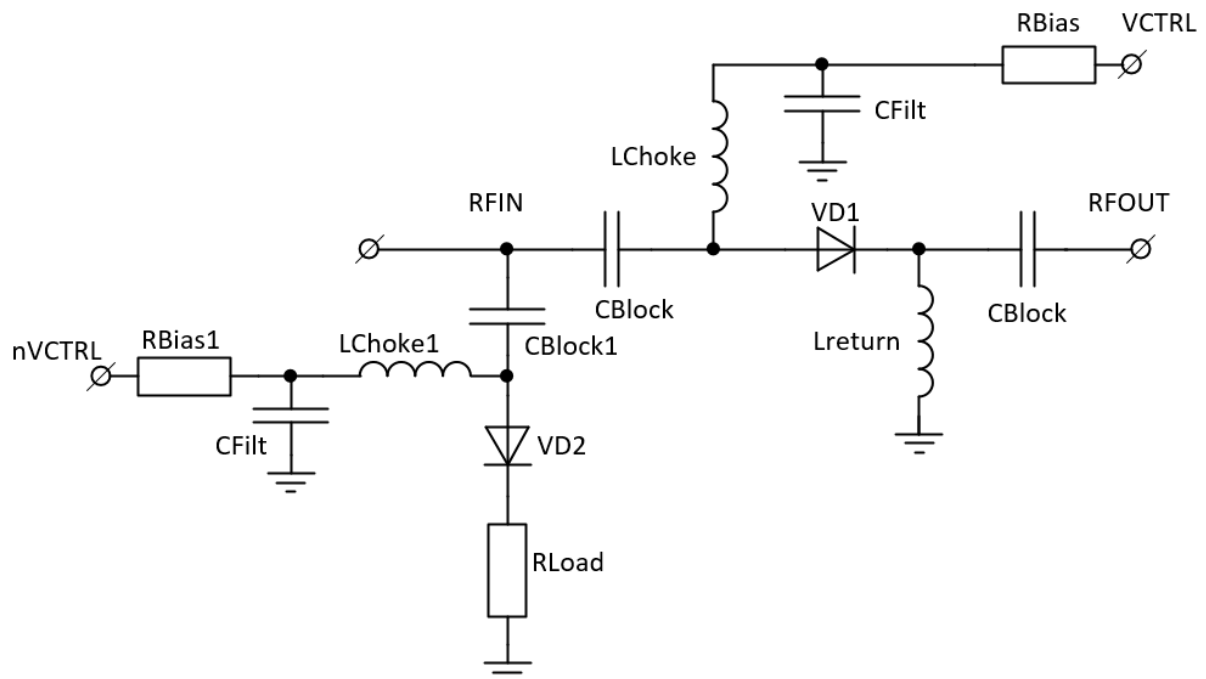
Данная схема SPST-ключа может выступать как одиночное звено для построения комбинированных схем, как например для SPDT-ключа ниже.



В этой схеме SPDT-ключа общим выходом является  $RFC$ , а подключенным в зависимости от подаваемых управляемых напряжений  $VCTRL1$  и  $VCTRL2$ , является выход  $RF1$  или  $RF2$ . Если на  $VCTRL1$  подать

высокое напряжение, на VCTRL2 0 В или отрицательное – PIN-диод VD1 открыт, VD2 – закрыт и с общим выходом RFC соединен выход RF1. Аналогично, если на VCTRL1 подать 0 В или отрицательное запирающее напряжение, а на VCTRL2 высокое, то PIN-диод VD1 закрыт, VD2 открыт и с общим выходом RFC соединен выход RF2. Одновременное закрытие обоих PIN-диодов приводит к тому, что вход RFC ни к чему не подключен. А вот одновременное открытие обоих PIN-диодов как правило не используется, для ключа это не является штатным режимом работы.

В схемах выше в разомкнутом выходе будет коэффициент отражения близок к 1. Это не всегда приемлемо, некоторые устройства требуют, чтобы разомкнутый канал был неотражающим. Чтобы этого достичь, можно модифицировать схему, добавив еще одно звено, которое будет замыкать отключенный канал в нагрузку. На примере показана модифицированная схема неотражающего SPST-ключа (показано добавочное звено только для входа RFIN, для выхода RFOUT все аналогично).



Данная схема по сути очень похожа на SPDT-ключ, где второй выход подключен на нагрузку RLoad. PIN-диоды VD1 и VD2 в данной схеме должны включаться в противофазе. Также, т.к. катод VD2 подключен в нагрузку RLoad без блокировочного конденсатора, то ток в прямом направлении на диоде VD2 задается уже парой RBias1+RLoad, это нужно учитывать.

## Методика выполнения работы

### Создание и первоначальная настройка проекта

#### Lab1

При создании проекта рекомендовано использовать стандартную технологию, что и в остальных проектах цикла, в том числе:

- В качестве библиотеки слоев необходимо дополнительно к библиотеке назначения слоев схемы «ads\_schematic\_layer» выбрать «ads\_standard\_layers» (назначение топологических слоев);

- единицы длин проекта – мм, с разрешением 10000 точек/мм.

Ключ мы будем проектировать на основе PIN-диода BAR90-02EL Infineon [7]. Для него производителем предоставлены S-параметры, измеренные в нескольких смещениях, как прямых, так и обратных [9].


Производитель выложил несколько примеров спроектированных ключей [10]. При проектировании будем ориентироваться на заявленные производителем характеристики ключей.

Диапазон частот, для которых будет спроектирован ключ – от 2 до 3 ГГц.

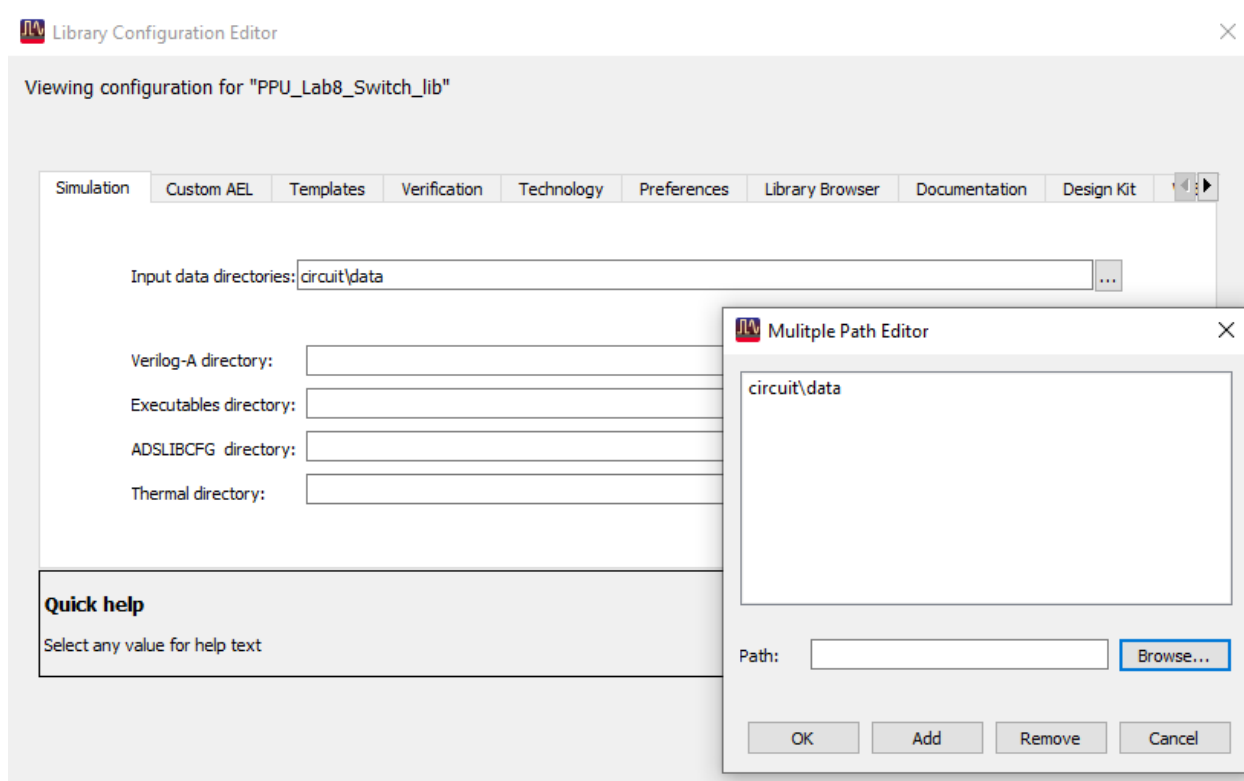
#### Создание компонента-хранилища S-параметров PIN-диода

Т.к. при проектировании будем опираться на файлы S-параметров, измеренных для различных смещений, то удобно их скомпоновать в виде компонента-хранилища файлов S-параметров. В этом компоненте при изменении параметра смещения (пусть будет «VDBias») будет подгружаться соответствующий файл S-параметров.

Для переносимости проекта удобно использовать относительные пути для подгружаемых файлов S-параметров, если хранить эти файлы в подпапке data в папке проекта \_wrk. Однако, если необходимо использовать проект как библиотеку с подключением в другой проект, то подпапка data не будет видна другому проекту. В этом случае правильнее хранить подгружаемые файлы в пределах подпапки библиотеки проекта \_lib.

 Чтобы компоненты видели эти модели, необходимо настроить библиотеку проекта. Для этого, в настройках библиотеки проекта (основное окно ADS, режим отображения Library View) необходимо по ПКМ – Configure library зайти в настройки библиотеки проекта. Затем на вкладке Simulation в поле Input data directories нужно добавить относительные пути (по отношению к подпапке \_lib), где будут храниться подгружаемые файлы

моделей. Рекомендовано подгружаемые файлы хранить в подпапке circuit\data или circuit\models аналогично настройкам по умолчанию при разработке библиотеки компонентов типа DesignKit. При первоначальной настройке проекта эти подпапки также необходимо будет создать.

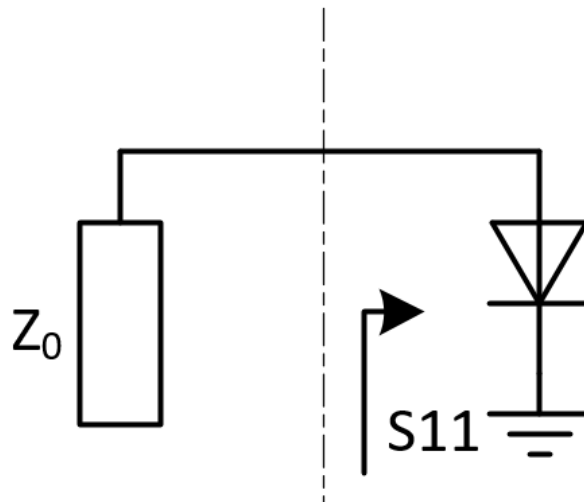


В эту папку разархивируем все файлы S-параметров для используемого PIN-диода BAR90-02EL [9].

DATA > ADS > ADS2023 > PPU_upd > PPU_Lab8_Switch_wrk > PPU_Lab8_Switch_lib > circuit > data				
Имя	Дата изменения	Тип	Размер	
BAR90-02EL_IAC_40mA.s1p	06.02.2019 21:33	Файл "S1P"	4 КБ	
BAR90-02EL_IAC_50mA.s1p	06.02.2019 21:33	Файл "S1P"	4 КБ	
BAR90-02EL_IAC_60mA.s1p	06.02.2019 21:33	Файл "S1P"	4 КБ	
BAR90-02EL_IAC_70mA.s1p	06.02.2019 21:33	Файл "S1P"	4 КБ	
BAR90-02EL_IAC_80mA.s1p	06.02.2019 21:33	Файл "S1P"	4 КБ	
BAR90-02EL_IAC_90mA.s1p	06.02.2019 21:33	Файл "S1P"	4 КБ	
BAR90-02EL_IAC_100mA.s1p	06.02.2019 21:33	Файл "S1P"	4 КБ	
BAR90-02EL_VCA_0V.s1p	06.02.2019 21:33	Файл "S1P"	4 КБ	
BAR90-02EL_VCA_1V.s1p	06.02.2019 21:33	Файл "S1P"	4 КБ	
BAR90-02EL_VCA_2V.s1p	06.02.2019 21:33	Файл "S1P"	4 КБ	
BAR90-02EL_VCA_3V.s1p	06.02.2019 21:33	Файл "S1P"	4 КБ	

Создадим ячейку BAR90-02EL\_SPAR, которая будет нашим компонентом-хранилищем. В ней создадим схематик, в котором разместим блок S1P и два пина (A и C). S-параметры PIN-диодов как правило снимаются в виде 2х-полюсника при подключении анод – смещение+ВЧ, катод – земля.

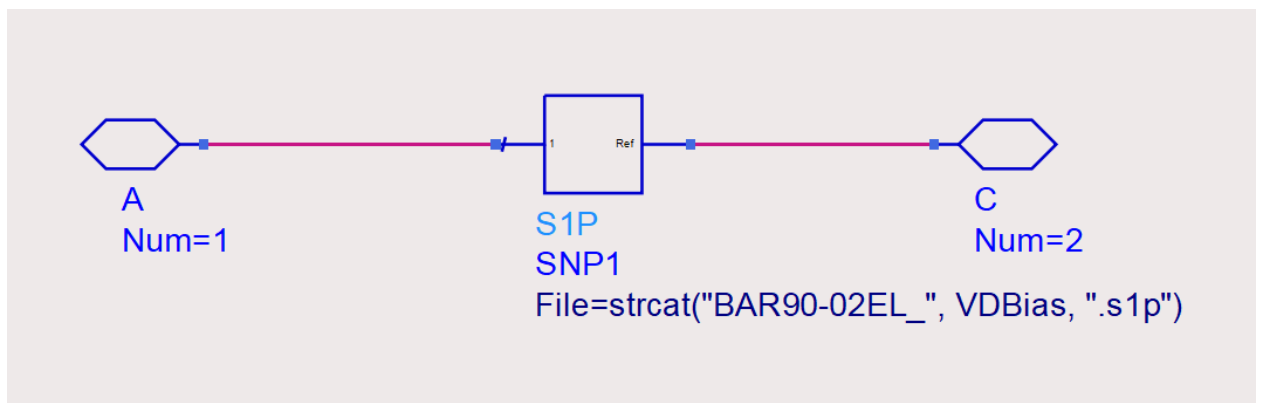




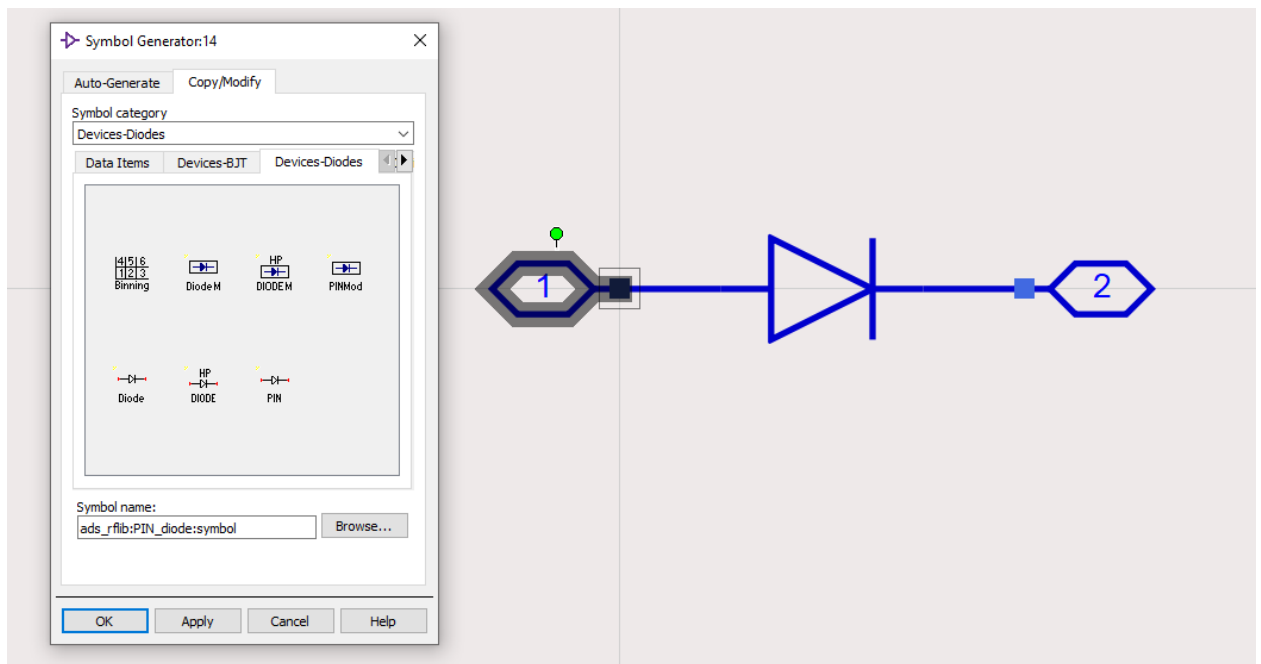
Соответственно, вход 1 у блока S1P – это анод, вход Ref – катод.

Определим, как будем собирать имя файла S-параметров. Видно, что имя файла можно представить в виде сборной строки с помощью функции `strcat("BAR90-02EL_", VDBias, ".s1p")`. Введем это в параметр File блока S1P.

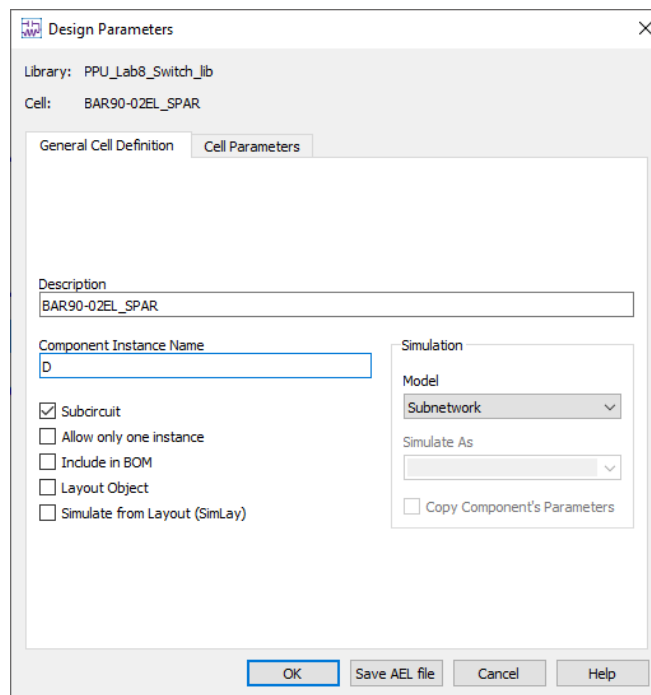
Внешний вид внутренности подсистемы BAR90-02EL\_SPAR будет выглядеть, как показано на рисунке ниже.



Далее создадим символ в ячейку BAR90-02EL\_SPAR. Чтобы не рисовать его с нуля, можно скопировать существующий символ `ads_rflib:PIN_diode`. Воспользуемся генератором символов. Единственное, необходимо подправить имена пинов анода и катода на «A» и «C» соответственно, аналогично подсистеме.



Теперь перейдем к настройке параметров компонента хранилища. Необходимо запустить команду из схематика File – Design Parameters. На вкладке General Cell Definitions укажем позобозначение по умолчанию как D (в поле Component Instance Name).

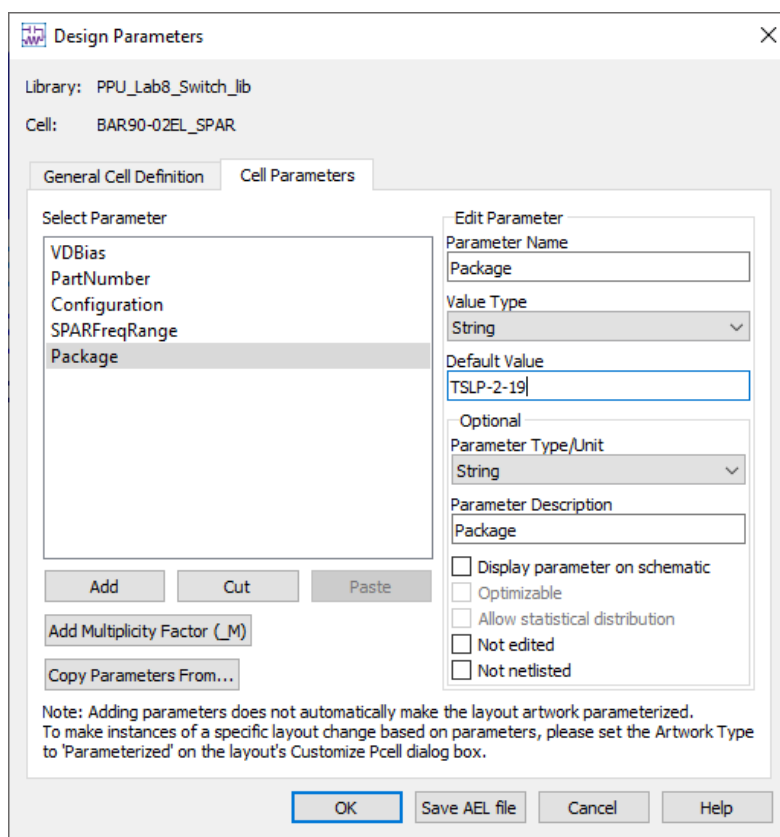


Затем перейдем на вкладку Cell Parameters и создадим несколько текстовых параметров (типа String) в соответствии с таблицей ниже. Значения PartNumber, Configuration и Package укажем на основании данных производителя [8]. Диапазон частот SPARFreqRange определим исходя из диапазона частот в файлах S-параметров [9].

**Таблица 1. Параметры компонента BAR90-02EL\_SPAR**

Имя параметра	Значение по умолчанию	Назначение
VDBias	IAC_0.01mA	Состояние текущего подгружаемого файла S-параметров
PartNumber	BAR9002ELE6327XTMA1	Обозначение компонента по номенклатуре производителя
Configuration	Single	Схема включения диода в корпусе
SPARFreqRange	From 300MHz To 12.9GHz	Частотный диапазон в файле S-параметров
Package	TSLP-2-19	Обозначение корпуса по номенклатуре производителя

Список параметров должен выглядеть приблизительно как показан ниже.



Однако, для перебора файлов S-параметров с помощью переменной VDBias нам необходимо использовать перечисление (enumeration) со списком файлов S-параметров. С помощью данного окна перечисление создать невозможно, необходимо будет править ael-код описания ячейки компонента.

Нажимаем на кнопку Save AEL file, чтобы в подпапке \_lib\BAR90-02EL\_SPAR появился файл itemdef.ael. Его мы и будем править.



Для сохранения переносимости проекта между системами, ADS может добавить в имя ячейки спецсимволы. Например, имя ячейки BAR90-02EL\_SPAR может измениться на «%B%A%R90#2d02%E%L\_%S%P%A%R».

Ниже и далее используется IDE VSCode [12] с установленным расширением Keysight AEL [13].

Чтобы создать перечисление, сначала необходимо создать именованные константы. Создавать их нужно до создания компонента - вызова функции `create_item()`

Именованные константы создаются с помощью функции

```
create_constant_form(name, label, attribute, netlistFormat, displayFormat);
```

где

`name` – имя именованной константы,

`label` – то, как значение константы будет отображаться в окне настроек компонента,

`attribute` – битовая маска, описывающая как можно обращаться с константой,

`netlistFormat` – то, как значение константы будет передаваться в нетлист при расчете схемы,

`displayFormat` – то, как будет отображаться константа при отображении на схеме.

Для PIN-диода BAR90-02EL производителем поставляются файлы S-параметров при прямом смещении IAC (нормируются по прямому току) в значениях от 0,1 мА до 1 мА с шагом 0,1 мА, затем с 1 мА до 10 мА с шагом 1 мА и от 10 мА до 100 мА с шагом 10 мА. И также доступны файлы при обратном смещении VCA (нормируются по обратному напряжению) от 0 В до 10 В с шагом 1 В и дополнительно 20 В.

Будем обозначать константы для прямого смещения в следующем формате (пример для тока 0,1 мА) "`constIAC0mA1`" с отображением в параметры компонента в виде строки "`IAC_0.1mA`". Для того, чтобы значение константы интерпретировалось как строка, это значение еще нужно обернуть в "\" как

показано ниже. Также не забудем именованные константы для обратного смещения в виде "VCA\_1V".

```
create_constant_form("constIAC0mA1", "IAC_0.1mA", 0, "\"IAC_0.1mA\"", "IAC_0.1mA");
create_constant_form("constIAC0mA2", "IAC_0.2mA", 0, "\"IAC_0.2mA\"", "IAC_0.2mA");
create_constant_form("constIAC0mA3", "IAC_0.3mA", 0, "\"IAC_0.3mA\"", "IAC_0.3mA");
create_constant_form("constIAC0mA4", "IAC_0.4mA", 0, "\"IAC_0.4mA\"", "IAC_0.4mA");
create_constant_form("constIAC0mA5", "IAC_0.5mA", 0, "\"IAC_0.5mA\"", "IAC_0.5mA");
create_constant_form("constIAC0mA6", "IAC_0.6mA", 0, "\"IAC_0.6mA\"", "IAC_0.6mA");
create_constant_form("constIAC0mA7", "IAC_0.7mA", 0, "\"IAC_0.7mA\"", "IAC_0.7mA");
create_constant_form("constIAC0mA8", "IAC_0.8mA", 0, "\"IAC_0.8mA\"", "IAC_0.8mA");
create_constant_form("constIAC0mA9", "IAC_0.9mA", 0, "\"IAC_0.9mA\"", "IAC_0.9mA");
create_constant_form("constIAC1mA", "IAC_1mA", 0, "\"IAC_1mA\"", "IAC_1mA");
create_constant_form("constIAC2mA", "IAC_2mA", 0, "\"IAC_2mA\"", "IAC_2mA");
create_constant_form("constIAC3mA", "IAC_3mA", 0, "\"IAC_3mA\"", "IAC_3mA");
create_constant_form("constIAC4mA", "IAC_4mA", 0, "\"IAC_4mA\"", "IAC_4mA");
create_constant_form("constIAC5mA", "IAC_5mA", 0, "\"IAC_5mA\"", "IAC_5mA");
create_constant_form("constIAC6mA", "IAC_6mA", 0, "\"IAC_6mA\"", "IAC_6mA");
create_constant_form("constIAC7mA", "IAC_7mA", 0, "\"IAC_7mA\"", "IAC_7mA");
create_constant_form("constIAC8mA", "IAC_8mA", 0, "\"IAC_8mA\"", "IAC_8mA");
create_constant_form("constIAC9mA", "IAC_9mA", 0, "\"IAC_9mA\"", "IAC_9mA");
create_constant_form("constIAC10mA", "IAC_10mA", 0, "\"IAC_10mA\"", "IAC_10mA");
create_constant_form("constIAC20mA", "IAC_20mA", 0, "\"IAC_20mA\"", "IAC_20mA");
create_constant_form("constIAC30mA", "IAC_30mA", 0, "\"IAC_30mA\"", "IAC_30mA");
create_constant_form("constIAC40mA", "IAC_40mA", 0, "\"IAC_40mA\"", "IAC_40mA");
create_constant_form("constIAC50mA", "IAC_50mA", 0, "\"IAC_50mA\"", "IAC_50mA");
create_constant_form("constIAC60mA", "IAC_60mA", 0, "\"IAC_60mA\"", "IAC_60mA");
create_constant_form("constIAC70mA", "IAC_70mA", 0, "\"IAC_70mA\"", "IAC_70mA");
create_constant_form("constIAC80mA", "IAC_80mA", 0, "\"IAC_80mA\"", "IAC_80mA");
create_constant_form("constIAC90mA", "IAC_90mA", 0, "\"IAC_90mA\"", "IAC_90mA");
create_constant_form("constIAC100mA", "IAC_100mA", 0, "\"IAC_100mA\"", "IAC_100mA");
create_constant_form("constVCA0V", "VCA_0V", 0, "\"VCA_0V\"", "VCA_0V");
create_constant_form("constVCA1V", "VCA_1V", 0, "\"VCA_1V\"", "VCA_1V");
create_constant_form("constVCA2V", "VCA_2V", 0, "\"VCA_2V\"", "VCA_2V");
create_constant_form("constVCA3V", "VCA_3V", 0, "\"VCA_3V\"", "VCA_3V");
create_constant_form("constVCA4V", "VCA_4V", 0, "\"VCA_4V\"", "VCA_4V");
create_constant_form("constVCA5V", "VCA_5V", 0, "\"VCA_5V\"", "VCA_5V");
create_constant_form("constVCA6V", "VCA_6V", 0, "\"VCA_6V\"", "VCA_6V");
create_constant_form("constVCA7V", "VCA_7V", 0, "\"VCA_7V\"", "VCA_7V");
create_constant_form("constVCA8V", "VCA_8V", 0, "\"VCA_8V\"", "VCA_8V");
create_constant_form("constVCA9V", "VCA_9V", 0, "\"VCA_9V\"", "VCA_9V");
create_constant_form("constVCA10V", "VCA_10V", 0, "\"VCA_10V\"", "VCA_10V");
create_constant_form("constVCA20V", "VCA_20V", 0, "\"VCA_20V\"", "VCA_20V");
```



При постепенном заполнении ael-файла в VSCode можно по команде Terminal – Run Build Task – ael:Compile active file скомпилировать ael-файл в atf-файл с автоматической проверкой корректности кода. При этом скомпилированный atf-файл автоматически в текущую сессию ADS не подгружается. Нужно либо переоткрыть проект в ADS, либо использовать функцию `load()` в командном окне ADS для загрузки в текущую сессию.

Далее, после создания констант их необходимо объединить в перечисление. Это делается с помощью функции

```
create_form_set(name, formName1, formName2,... formNameN);
```

где

`name` – имя перечисления,

`formName1.. formNameN` – имена констант.

Соберем все созданные выше константы в перечисление `"enumStateBAR90_02EL"`.

```
create_form_set("enumStateBAR90_02EL", "constIAC0mA1", "constIAC0mA2",  
"constIAC0mA3", "constIAC0mA4", "constIAC0mA5", "constIAC0mA6", "constIAC0mA7",  
"constIAC0mA8", "constIAC0mA9", "constIAC1mA", "constIAC2mA", "constIAC3mA",  
"constIAC4mA", "constIAC5mA", "constIAC6mA", "constIAC7mA", "constIAC8mA",  
"constIAC9mA", "constIAC10mA", "constIAC20mA", "constIAC30mA", "constIAC40mA",  
"constIAC50mA", "constIAC60mA", "constIAC70mA", "constIAC80mA", "constIAC90mA",  
"constIAC100mA", "constVCA0V", "constVCA1V", "constVCA2V", "constVCA3V",  
"constVCA4V", "constVCA5V", "constVCA6V", "constVCA7V", "constVCA8V",  
"constVCA9V", "constVCA10V", "constVCA20V");
```

Далее необходимо подправить создание параметров в виде вызовов `create_parm()` в составе аргументов функции `create_item()`.

Основную часть вызова функции `create_item()` мы править не будем, иначе очень легко подпортить создание компонента.

Функция создания параметров компонента имеет следующий синтаксис

```
create_parm(name, label, attrib, formSet, unitCode, defaultValue[, cbList]);
```

где

`name` – имя параметра,

`label` – описание параметра,

`attrib` – битовая маска, описывающая тип параметра,

`formSet` – тип данных, описывающий какие значения может принимать параметр, может быть или из стандартного типа данных `"StdFormSet"` или из созданного самостоятельно, `"enumStateBAR90_02EL"`,

`unitCode` – код размерности единиц,

`defaultValue` – значение по умолчанию, создается с помощью дополнительной функции `prm()`;

`cbList` – список колбэков, вызов которых можно привязать при изменении параметра, необязательный параметр.

Переопределим строку с созданием параметра `VDBias`. Имя и описание менять не будем. Т.к. это перечисление, то никаких разрешений на него давать не будем, битовая маска будет 0. Далее указываем созданное перечисление `"enumStateBAR90_02EL"`. Размерность единицы будет безразмерная -1. В значение по умолчанию выберем первую константу `"constIAC0mA1"`. Колбеков у нас нет, их привязывать не будем

```
create_parm("VDBias", "VDBias", 0, "enumStateBAR90_02EL", -1, prm("constIAC0mA1")),
```

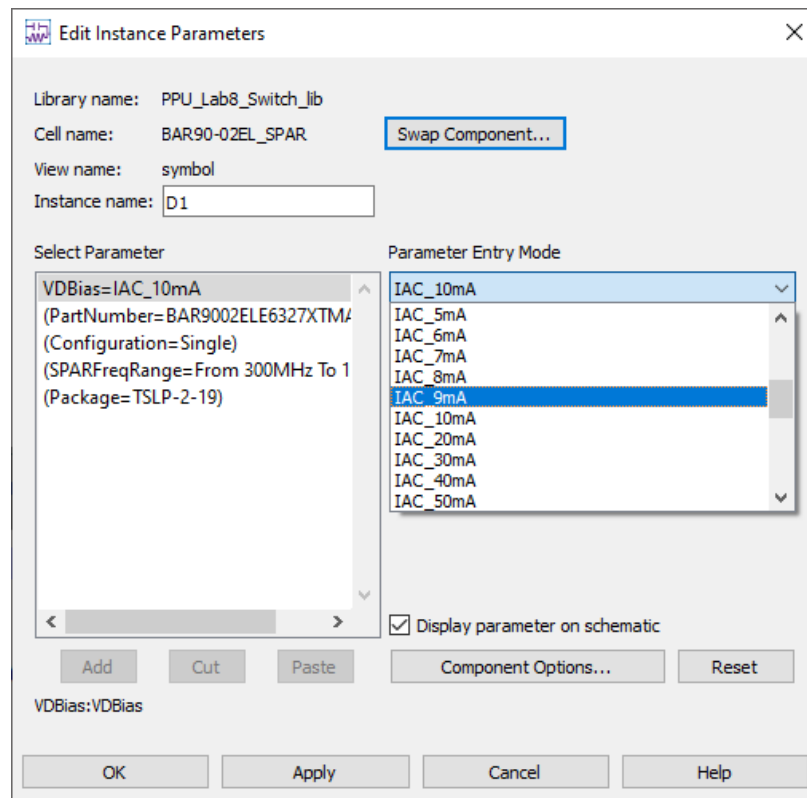
Остальные параметры выступают в виде строковых информационных. Им только подправим битовую маску на значение 577 (или `PARAM_NOT_EDITED | PARAM_NOT_NETLISTED | PARAM_NO_DISPLAY`).

```
create_parm("PartNumber", "PartNumber", 577, "StdFormSet", -2, prm("StdForm", "BAR9002ELE6327XTMA1")),
create_parm("Configuration", "Configuration", 577, "StdFormSet", -2, prm("StdForm", "Single")),
create_parm("SPARFreqRange", "SPARFreqRange", 577, "StdFormSet", -2, prm("StdForm", "From 300MHz To 12.9GHz")),
create_parm("Package", "Package", 577, "StdFormSet", -2, prm("StdForm", "TSLP-2-19"));
```

Общий вид вызова `create_item()` должен выглядеть приблизительно следующим образом:

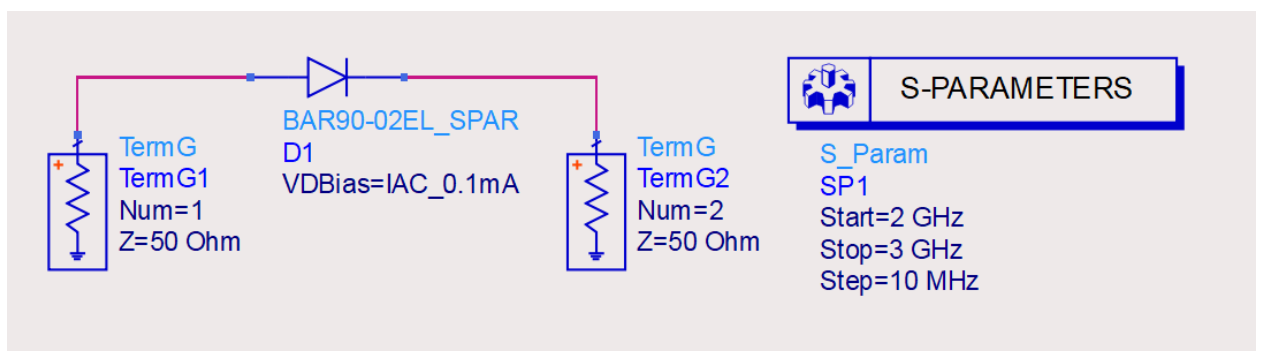
```
create_item("BAR90-02EL_SPAR", "BAR90-02EL_SPAR", "D", 16, -1, NULL, "Component Parameters", NULL, "%43?global %; %d: %t %# %44?0%:%31?%C%:_net%c%; %e %b%r%8?%29?%:%30?%p %:%k%?[ %1i ]%; =%p %; %; %; %e%e", "BAR90-02EL_SPAR", "%t%b%r%38?%:\n%30?%s%:%k%?[ %1i ]%; =%s%; %; %e%e%", "", 3, NULL, 0, create_parm("VDBias", "VDBias", 0, "enumStateBAR90_02EL", -1, prm("constIAC0mA1")), create_parm("PartNumber", "PartNumber", 577, "StdFormSet", -2, prm("StdForm", "BAR9002ELE6327XTMA1")), create_parm("Configuration", "Configuration", 577, "StdFormSet", -2, prm("StdForm", "Single")), create_parm("SPARFreqRange", "SPARFreqRange", 577, "StdFormSet", -2, prm("StdForm", "From 300MHz To 12.9GHz")), create_parm("Package", "Package", 577, "StdFormSet", -2, prm("StdForm", "TSLP-2-19"));
```

Если все создано правильно, то теперь у нас готов компонент `BAR90-02EL_SPAR` как хранилище S-параметров. При размещении его в другие схемы можно в выпадающем списке выбирать текущее смещение и в симуляцию будет автоматически подгружаться необходимый файл S-параметров.



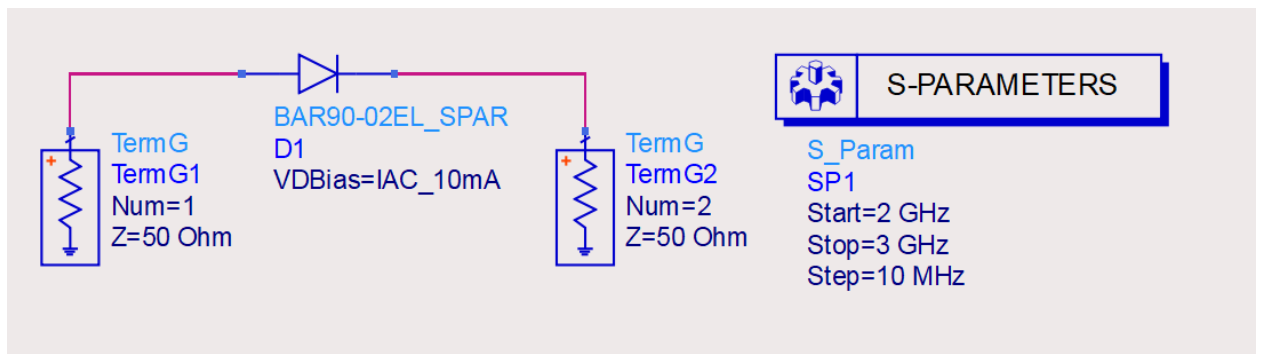
### Выбор режимов работы pin-диода

При проектировании ключа необходимо выбрать, в каких режимах мы будем использовать диод для открытия и закрытия канала. Соберем схему SW\_ChooseBias для оценки S-параметров PIN-диода при различных смещениях. Частотный диапазон выберем весь проектируемый от 2 до 3 ГГц.

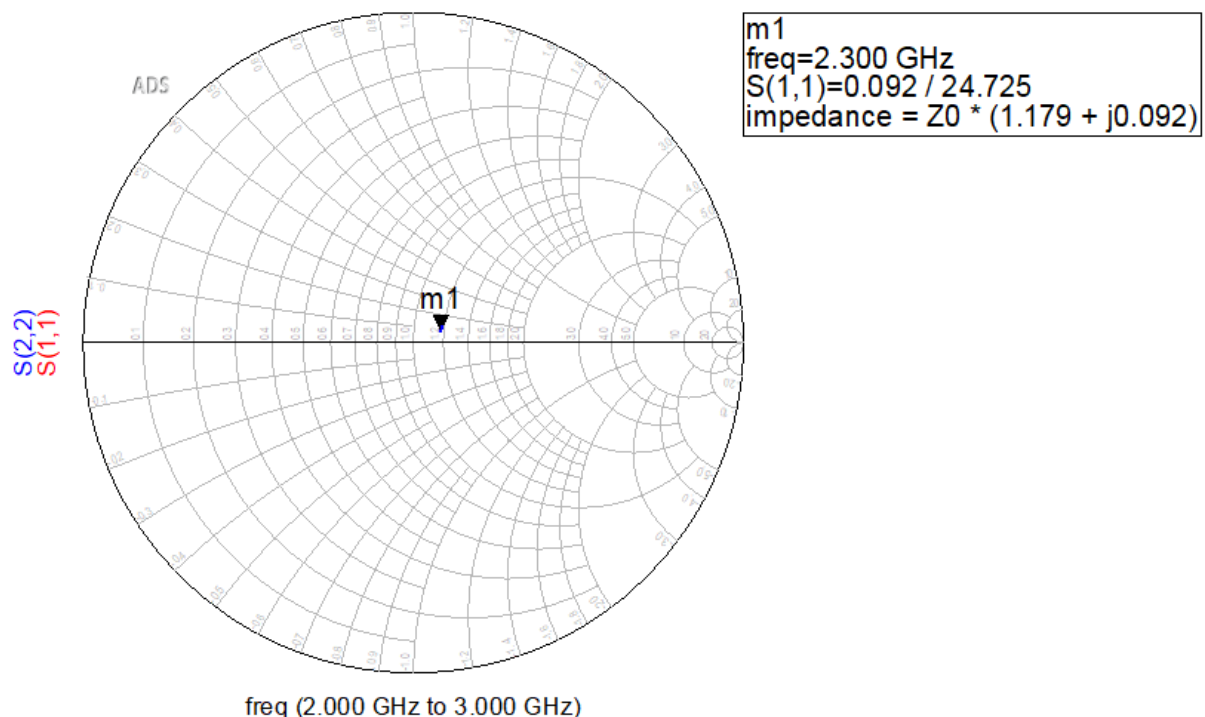


Сначала выберем прямой режим, когда PIN-диод открыт. В открытом состоянии нам желательно иметь как можно меньшие коэффициенты отражения. Переберем несколько состояний в прямом направлении. Прямое смещение  $I_F = 10$  мА является удобным. Остановимся на нем.



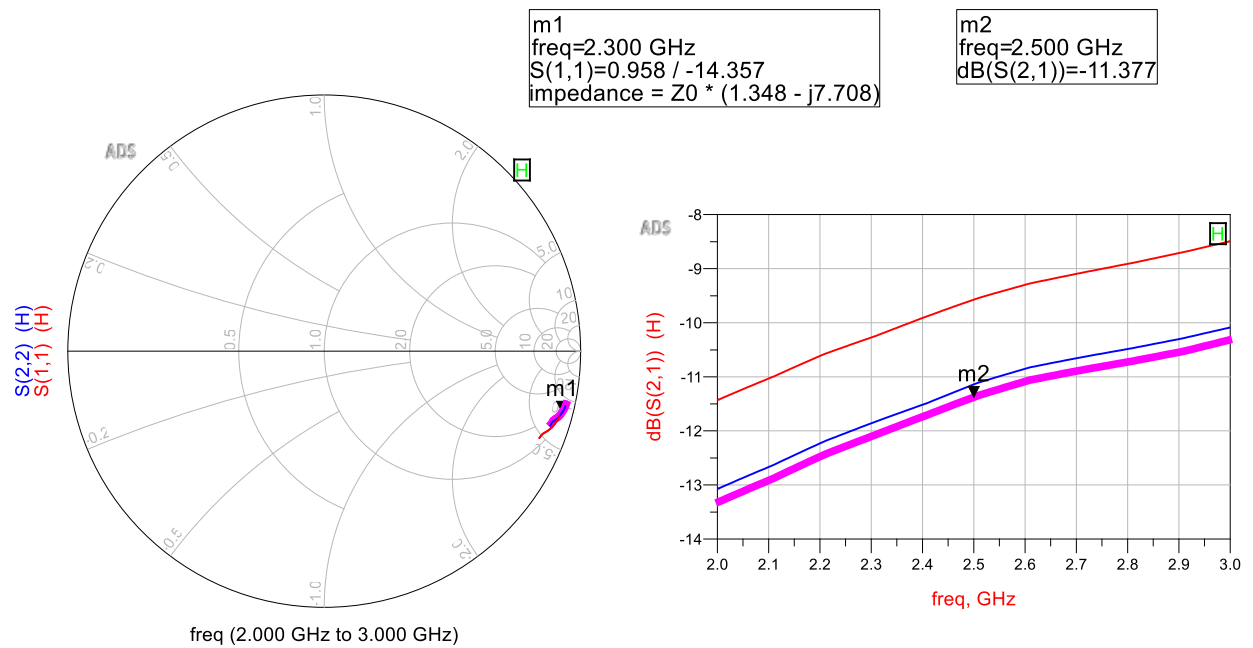


Коэффициент отражения минимален.



Далее выберем обратное смещение VR, когда PIN-диод закрыт. Рассматривать будем смещения 0 В, 3 В и 5 В как часто встречающиеся управляющие номиналы. Смотреть будем по положению S11 и коэффициенту передачи S21.

В схеме были последовательно устанавливаются PIN-диоду состояния VCA\_0V, VCA\_3V и VCA\_5V. В области графиков включен режим истории History – On. В этом режиме в областях графиков появляется метка «Н», цвета новых добавленных линий идут по стандартному цветовому порядку (первая – красная, вторая – синяя, третья – фиолетовая и т.д.), графики последнего состояния отмечаются жирным.



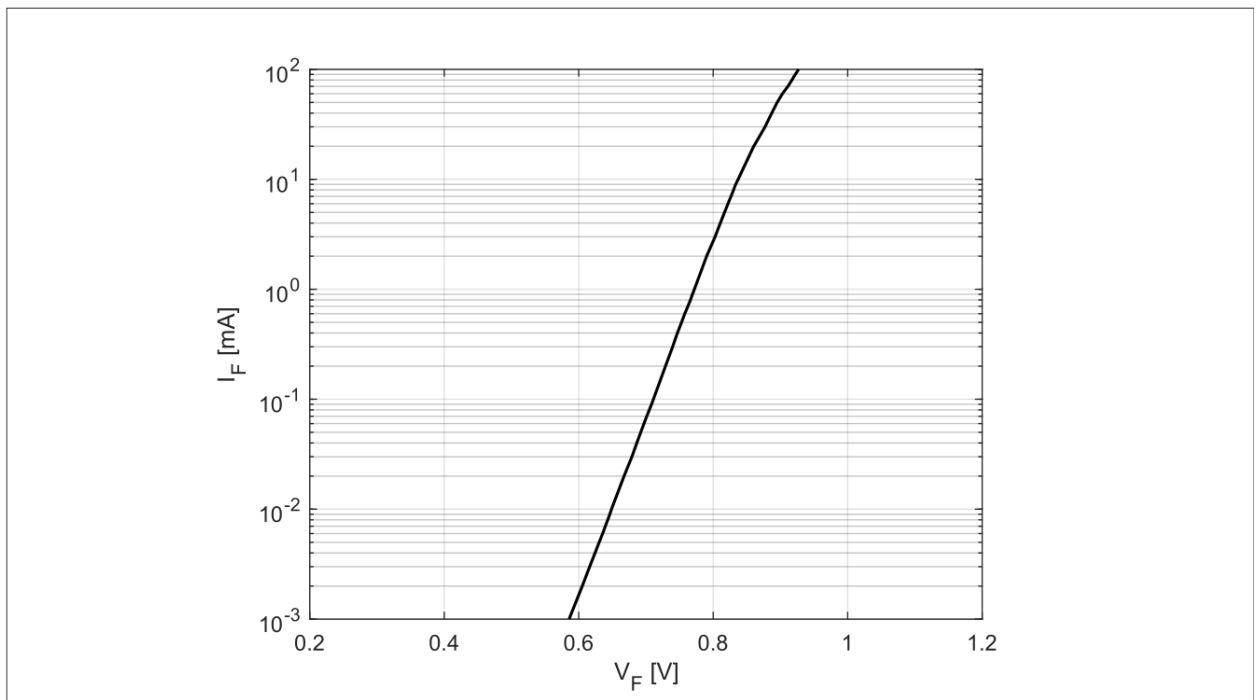
Видно, что влияние обратного смещения на коэффициент  $S_{11}$  минимально и он остается близким к 0,95 с небольшим поворотом фазы.

А вот по  $S_{21}$  влияние более значительное. Использование 0 В дает меньшее запираение. При этом, между обратными смещениями в 3 В (синяя линия) и 5 В (фиолетовая линия) особой разницы нет. Выбирать без необходимости большие по номиналу напряжения не стоит. Остановимся на обратном смещении  $VR = 3$  В для закрытия диода. Таким образом, для удобства управления и построения симметричной схемы нам предварительно удобно остановиться на управляющих напряжениях  $VCTRL = +3/-3$  В.

Надо учесть, что данный анализ проведен в линейном режиме, только на основании S-параметров. А вот с точки зрения нелинейных параметров ( $P_{1dB}$ ,  $IP3$ ,  $IMD$  и пр.) выбор управляющих режимов может оказаться не настолько очевидным.

### Расчет токоограничивающего резистора

Мы выбрали режим прямого смещения на  $IF = 10$  мА. На основании документации производителя [8] оценим, какому прямому падению напряжения  $V_F$  этот ток соответствует.



**Figure 1 Forward current  $I_F$  vs. forward voltage  $V_F$**

По графику получается прямое падение напряжения  $V_F \approx 0,83$  В.

Ранее мы выбрали управляющее напряжение  $V_{CTRL} = 3/-3$  В. Это означает, что номинал токоограничивающего резистора  $R$  для открытого режима  $V_{CTRL\_ON} = 3$  В получается равным

$$R = \frac{V_{Ctrl\_ON} - V_F}{I_F} = 217 \approx 220 \text{ Ом}.$$

Нужно помнить, что данный расчет получен с округлением, а также ВАХ конкретного PIN-диода может немного гулять от партии к партии. Это значит, что надо быть готовым к регулировке изготовленного ключа, в том числе через подбор точного номинала резистора при контроле управляющего тока в открытом состоянии PIN-диода.

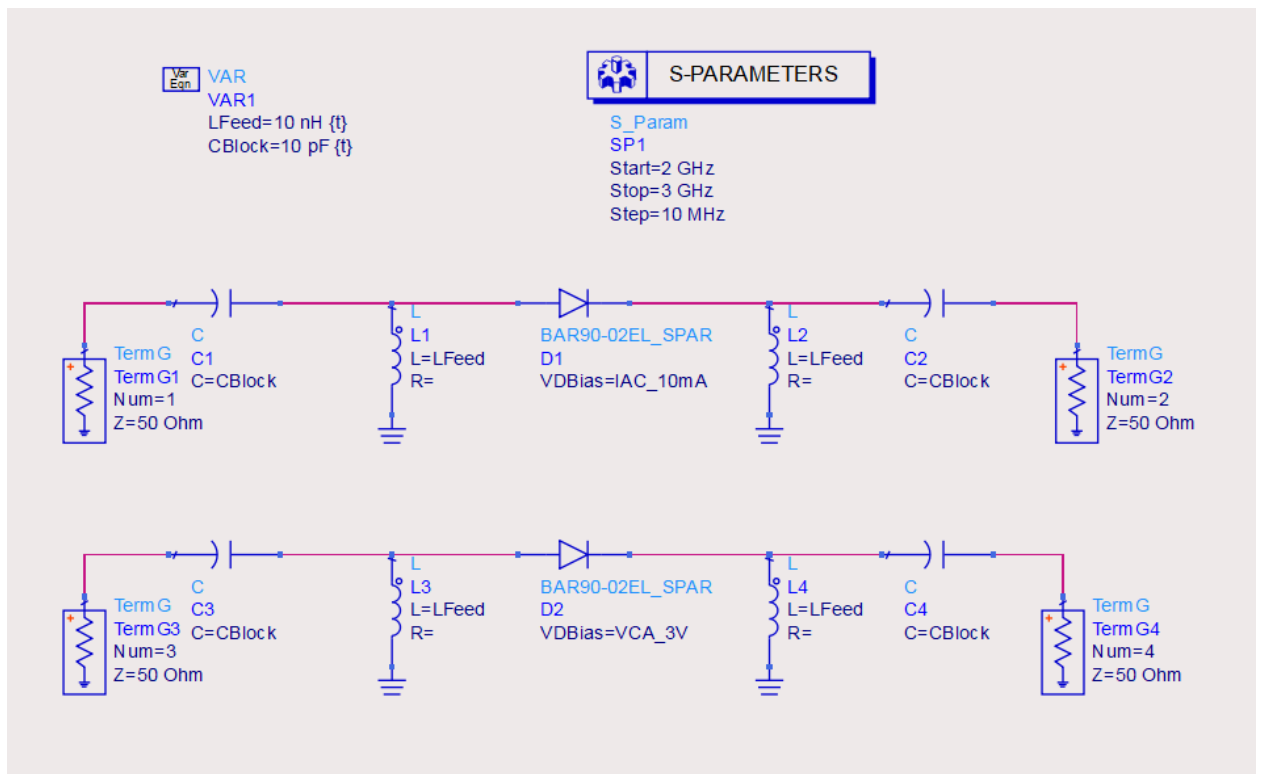
### Проектирование SPST-ключа

Исходя из выбранных смещений, спроектируем SPST-ключ (схема SW\_SPST). Также этот ключ будет выступать в виде одиночного звена в последующей работе. Используемая схема SPST-ключа построена на последовательном включении PIN-диода.

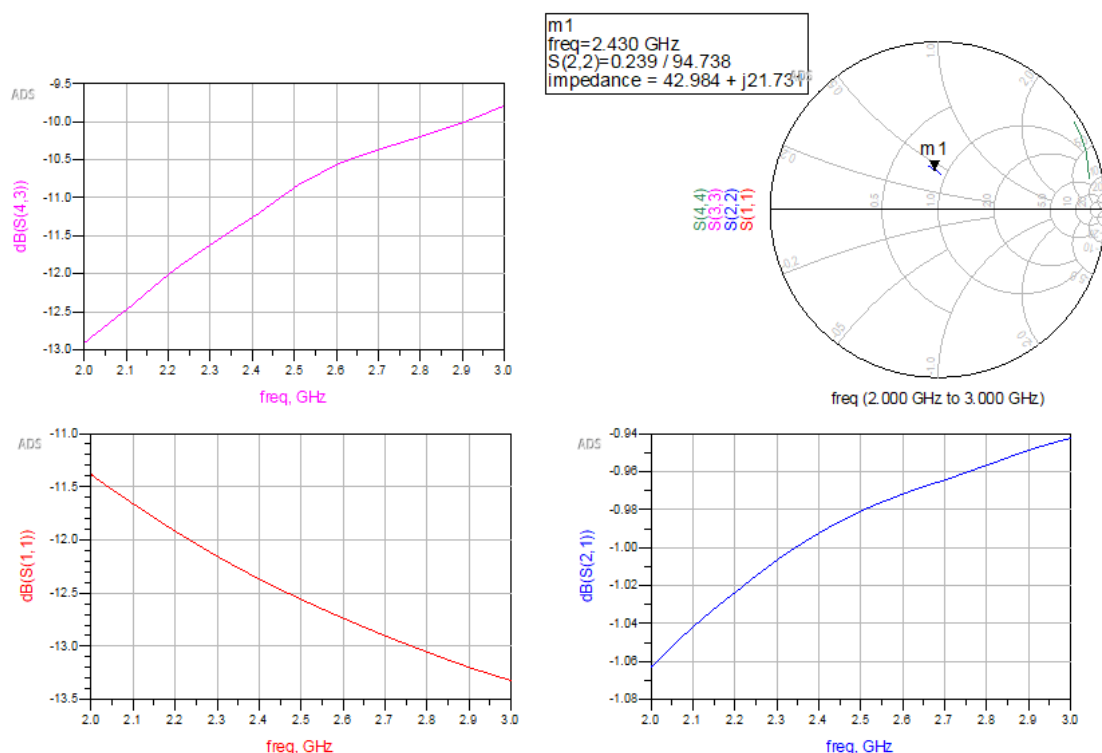
При подборе LC-компонентов надо будет оценивать открытый (замкнутый) и закрытый (разомкнутый) режимы одновременно. В открытом состоянии будем смотреть на коэффициент отражения и коэффициент передачи. В закрытом состоянии будем смотреть на запирающие.

Компонент-хранилище BAR90-02EL\_SPAR не поддерживает перебор состояний, поэтому сделаем двойную схему. Верхняя часть будет имитировать открытое состояние (порты 1-2, в S-параметры PIN-диода подгружено выбранное прямое смещение  $I_F = 10 \text{ mA}$ ), нижняя часть имитирует закрытое состояние (порты 3-4, в S-параметры PIN-диода подгружено выбранное обратное смещение  $V_R = 3 \text{ V}$ ).

Т.к. LC-цепи со стороны подачи управляющего напряжения похожи на выходную согласующую LC-цепь, то их проще изначально проектировать одинаковыми.



Исходя из нумерации портов в области графиков будем смотреть коэффициенты отражения на диаграмме Смита,  $S_{11}$  и  $S_{21}$  на прямоугольном графике для открытого канала и отдельно  $S_{43}$  для закрытого канала. Коэффициенты отражения открытого канала  $S_{11}$  и  $S_{22}$  будем стараться привести в состояние не хуже  $-15 \text{ dB}$ , потери в открытом состоянии  $S_{21}$  не хуже  $1 \text{ dB}$ , запираение в закрытом состоянии  $S_{43}$  не хуже  $10 \text{ dB}$ . При подборе LC-цепи по положению  $S_{11}$  и  $S_{22}$  на диаграмме Смита удобно следить, как двигаются  $S_{11}$  и  $S_{22}$ .



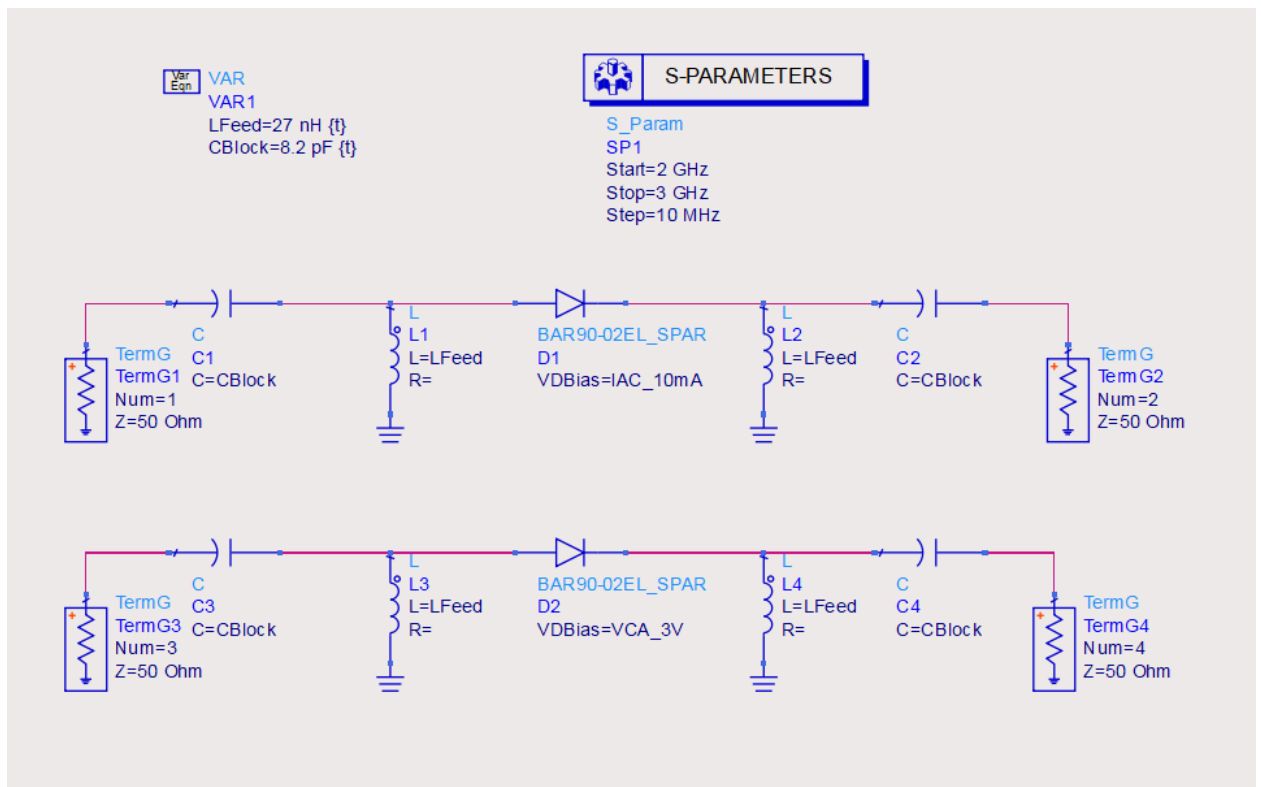
Подберем номиналы LC-цепи. Диапазон номиналов конденсаторов как правило получается где-то в диапазоне от 5 пФ до 100 пФ, диапазон номиналов индуктивностей где-то в диапазоне от 10 нГн до 100 нГн.

При подборе нужно помнить, что доступны не все произвольные номиналы дискретных компонентов, нужно выбирать из реально существующих. Для конденсаторов и индуктивностей можно считать, что доступны номиналы из ряда E24 (таблица 2). Дополнительно для конденсаторов близких к единицам пФ доступны номиналы с шагом чаще - от 0,5 пФ до 10 пФ с шагом 0,1 пФ и от 10 пФ до 20 пФ с шагом 1 пФ). И аналогично для индуктивностей близким к единицам нГн – массово доступны от 0,5 нГн до 10 нГн с шагом 0,1 нГн.

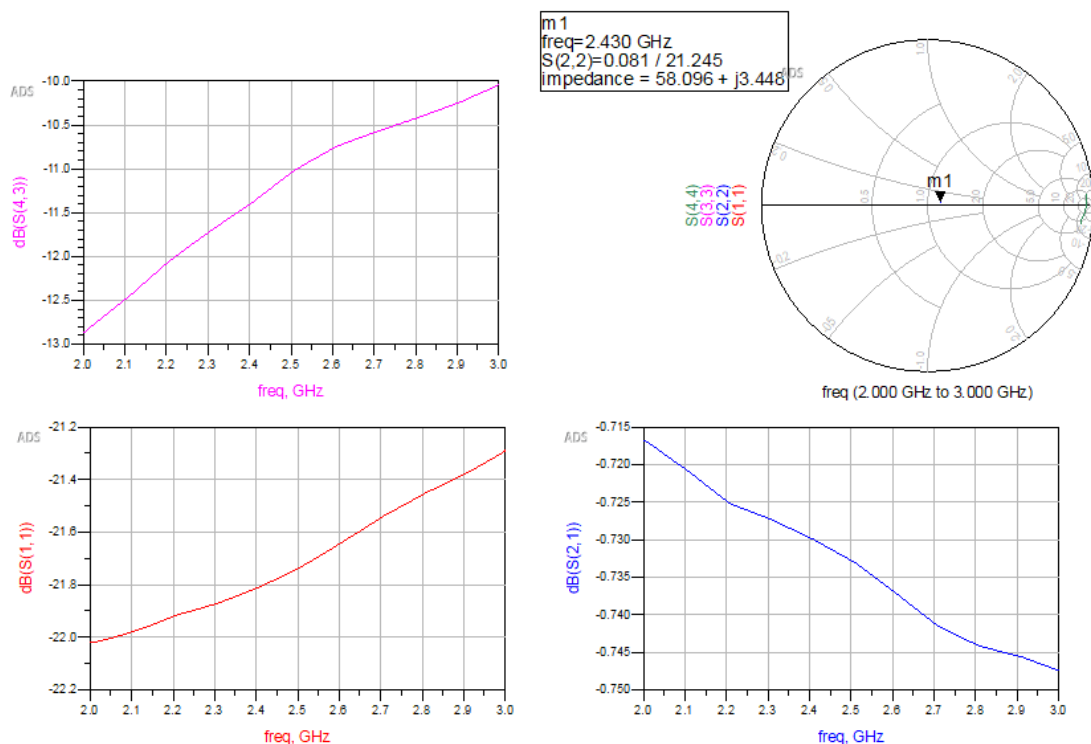
**Таблица 2. Ряд номиналов E24**

<b>Ряд E24</b>							
1,0	1,1	1,2	1,3	1,5	1,6	1,8	2,0
2,2	2,4	2,7	3,0	3,3	3,6	3,9	4,3
4,7	5,1	5,6	6,2	6,8	7,5	8,2	9,1

Остановимся на  $CB_{lock} = 8,2$  пФ и  $L_{Feed} = 27$  нГн.

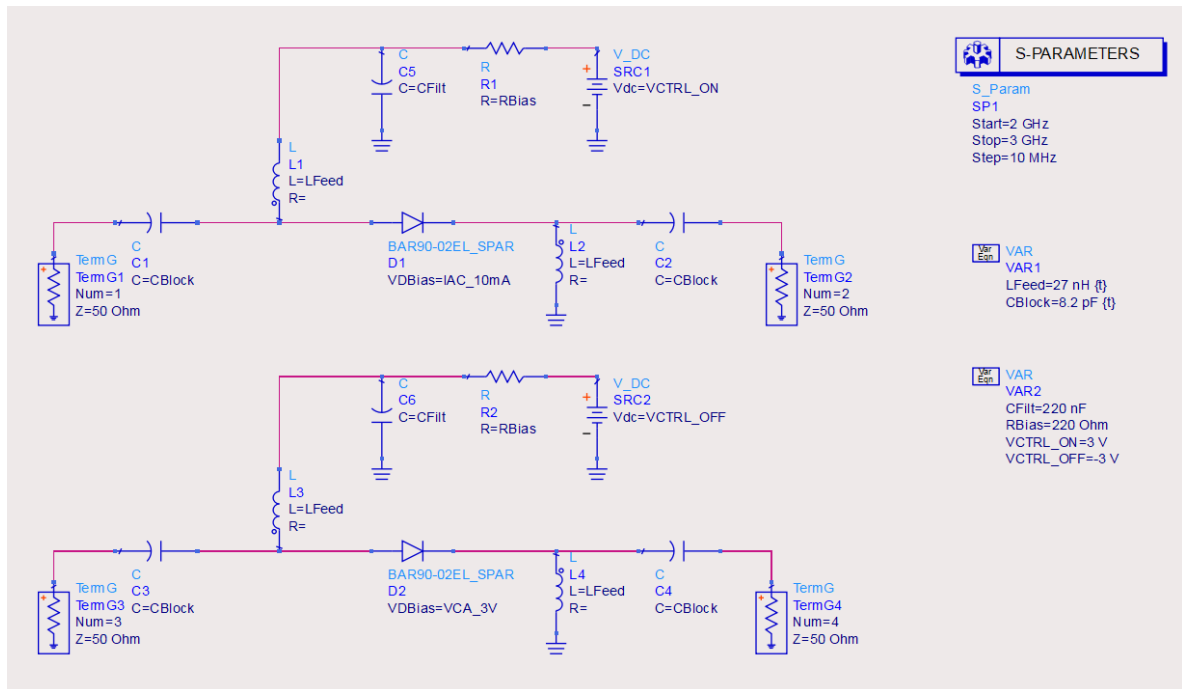


Данные значения дают в открытом канале потери порядка 0,7 дБ с коэффициентом отражения не хуже -21 дБ и запираение в закрытом канале не хуже 10 дБ.

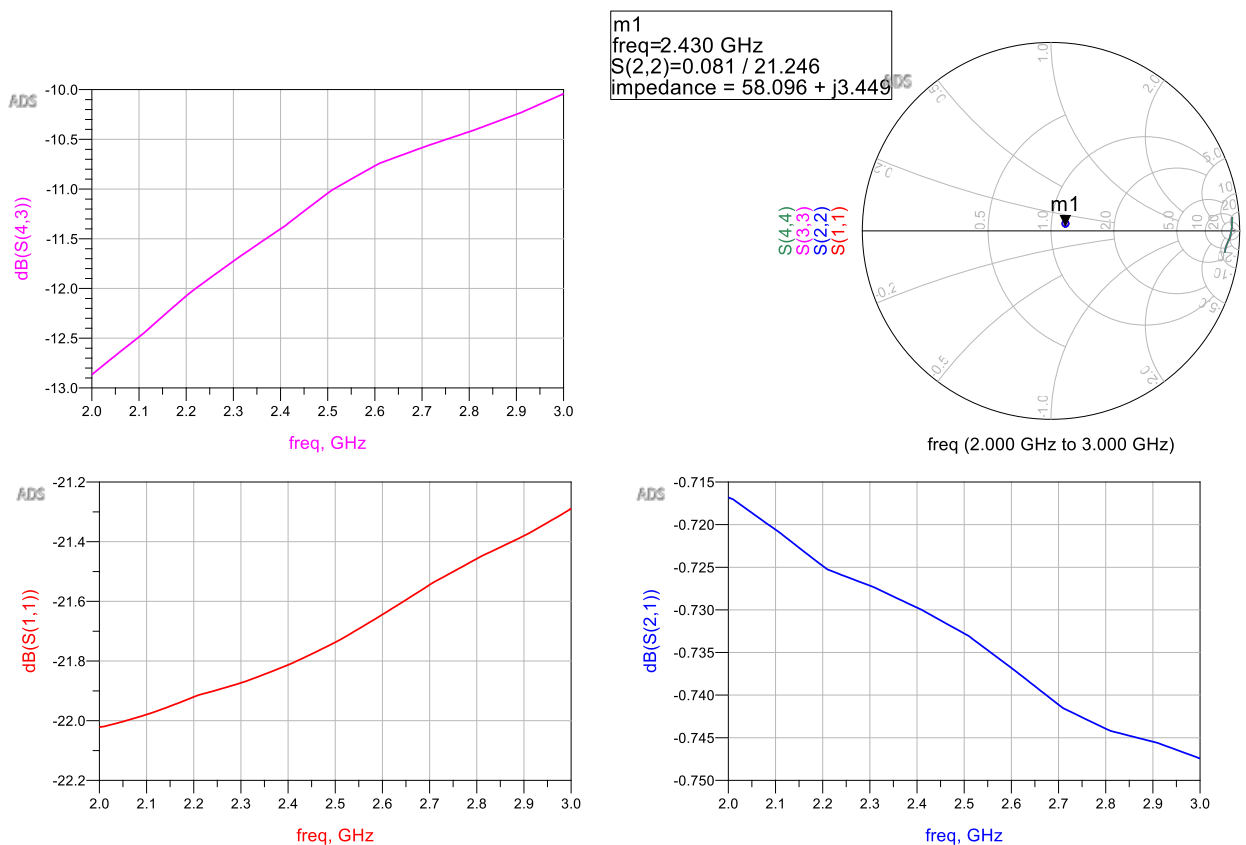


Доопределим схему до полного SPST-ключа. Индуктивности, через которые в анод PIN-диода подается смещение, необходимо оторвать от земли,

добавить к ним фильтрующий конденсатор CFilt (номиналом порядка 150..330 нФ), рассчитанный ранее токоограничивающий резистор RBias и источник постоянного напряжения, имитирующий управляющие сигналы VCTRL\_ON и VCTRL\_OFF.



При необходимости, подстроим номиналы LC-цепей и проконтролируем результаты.



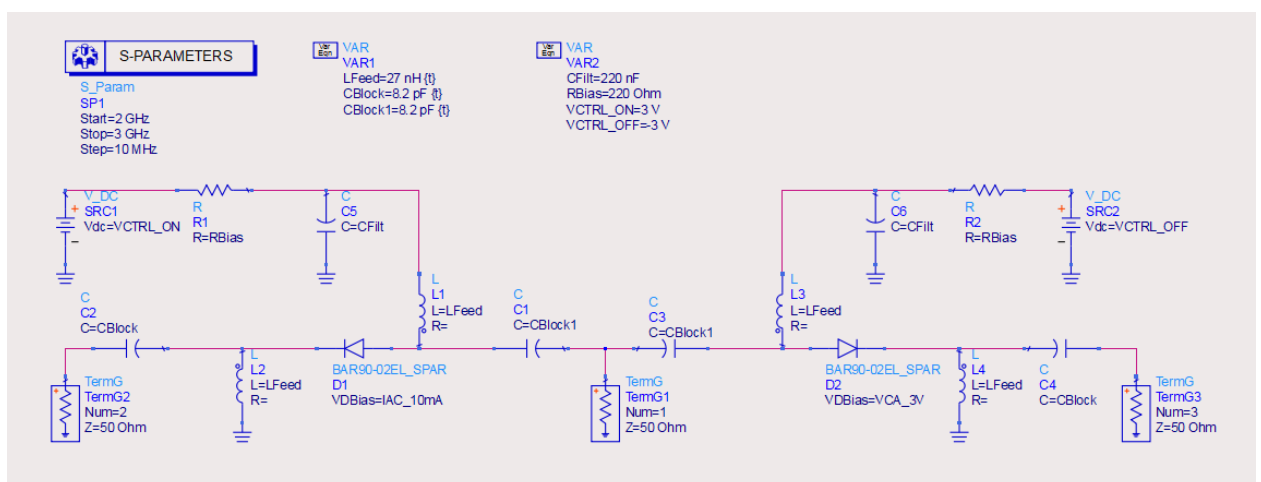
Модель SPST-ключа готова.

Нужно понимать, что невозможно промоделировать данную схему по постоянному току, т.к. в S-параметры не содержат поведение на нулевой частоте. Потому расчет  $R_{Bias}$  нет возможности проверить с помощью моделирования. Кроме того, S-параметры описывают линейное поведение, поэтому нет возможности оценить нелинейные свойства ( $P_{1dB}$ ,  $IP_3$ ,  $IMD$  и пр.). Кроме того, нет возможности промоделировать эту схему как управляемую с помощью изменения  $V_{CTRL} = +3\text{ В}$  или  $-3\text{ В}$ , т.к. переключение режима задается с помощью выбора соответствующего файла S-параметров. Для всего вышесказанного нужна достоверная нелинейная модель или испытания на макете.

### Проектирование SPDT-ключа

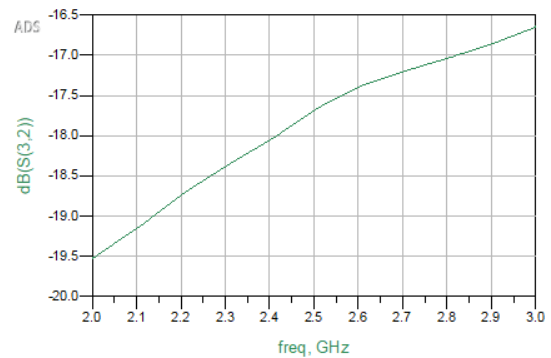
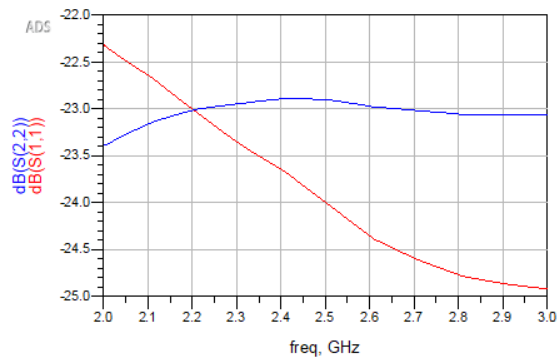
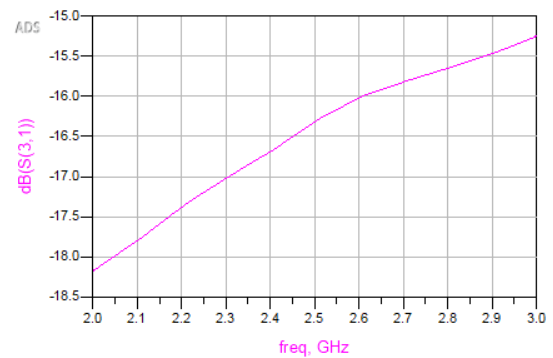
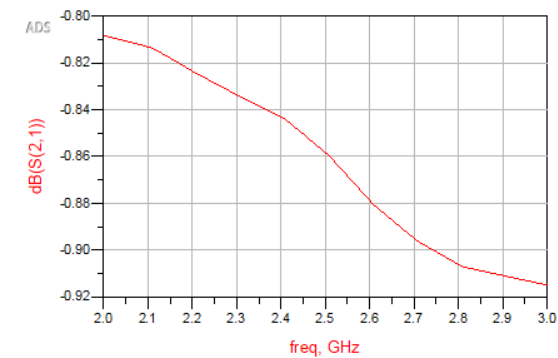
Воспользуемся спроектированной ранее схемой SPST-ключа как одиночным звеном для SPDT-ключа. Нас будет интересовать обычное использование такого ключа, когда общий вход RFC соединен с одним из выходов. Пусть это открытым будет RF1, тогда выход RF2 закрыт. Исходя из этого можно составить следующую схему (схема SW\_SPDT).

Порт 1 имитирует общий вход RFC, порт 2 имитирует выход RF1 (открытый), порт 3 имитирует выход RF2 (закрытый). Номинал блокировочных конденсаторов  $C_{Block1}$  на общем входе RFC вынесен в отдельную переменную, т.к. возможно его нужно будет подстраивать отдельно от  $C_{Block}$  в выходах RF1 и RF2.



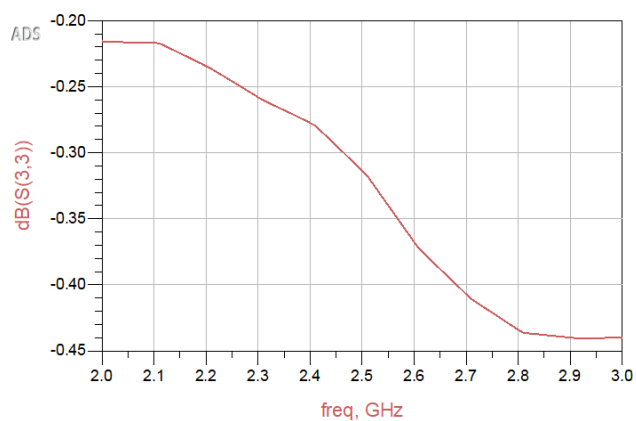
В области графиков интересует потери в открытом канале  $S_{21}$ , коэффициенты отражения общего  $S_{11}$  и открытого входов  $S_{22}$ , запираение в закрытый канал  $S_{31}$  и развязка между входами RF1 и RF2  $S_{32}$ .





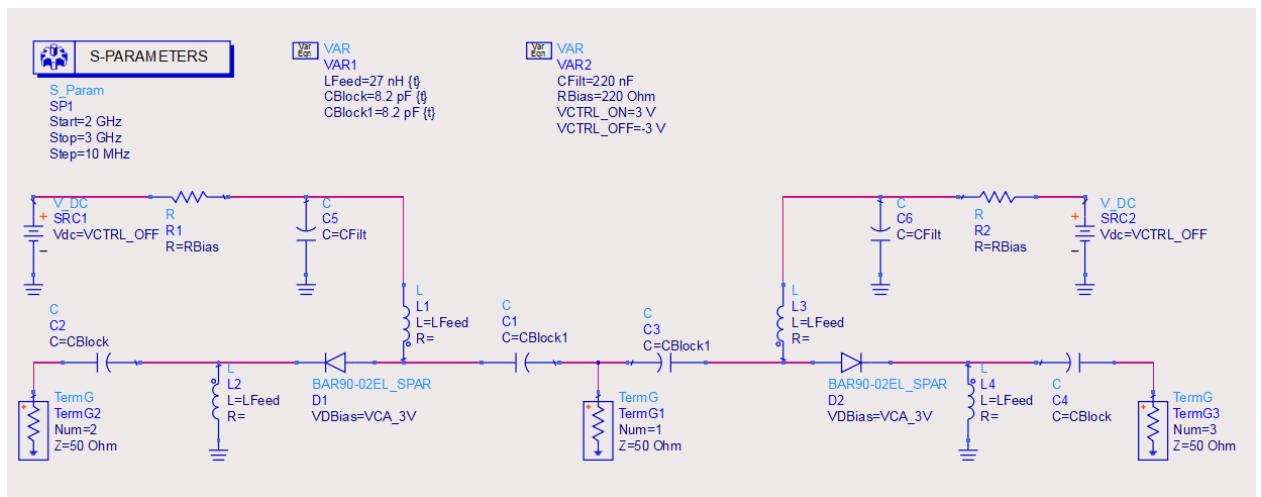
Значения получились приемлемые, кроме того, запирания чуть улучшились по отношению к SPST-ключу. Если потери и согласование в открытом канале ухудшается, то схему нужно немного подстроить, в первую очередь за счет CBlock1.

Также можно оценить, чему будет равен коэффициент отражения разомкнутого канала S33.

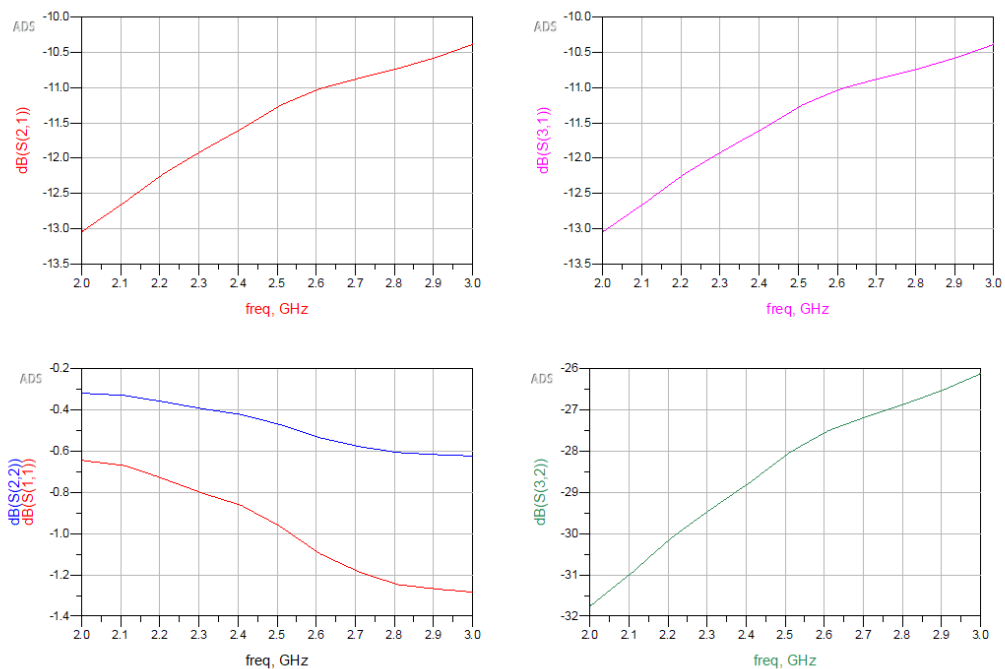


Как и ожидается, коэффициент отражения разомкнутого канала близок к 0 дБ.

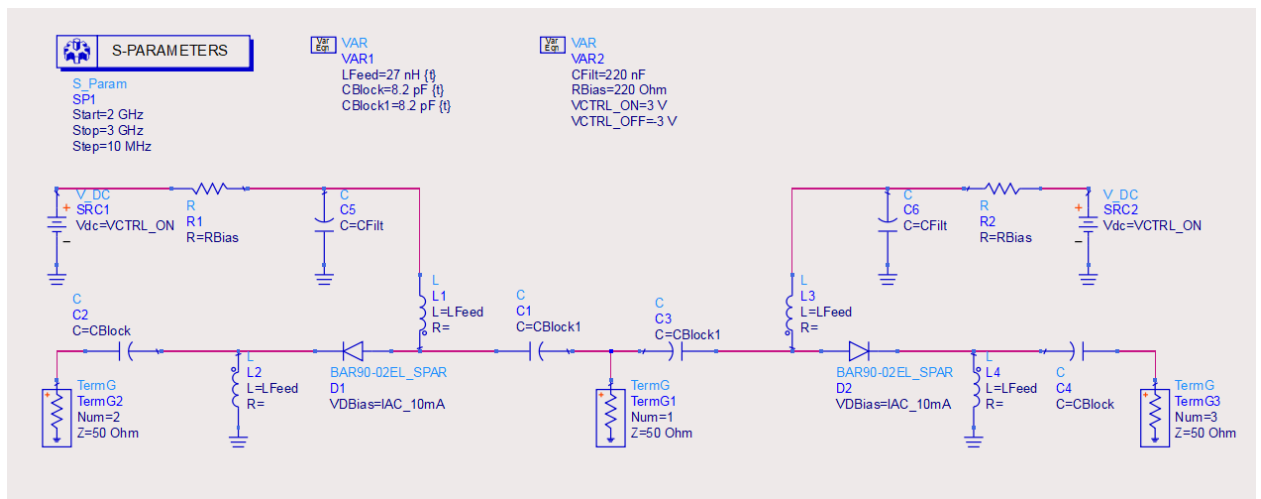
Можно изучить, как будет вести себя SPDT-ключ, если оба его канала разомкнуты.



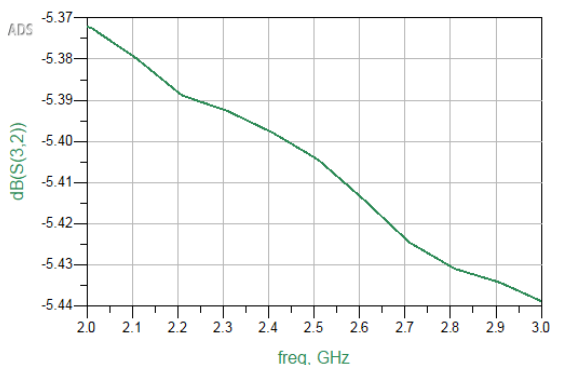
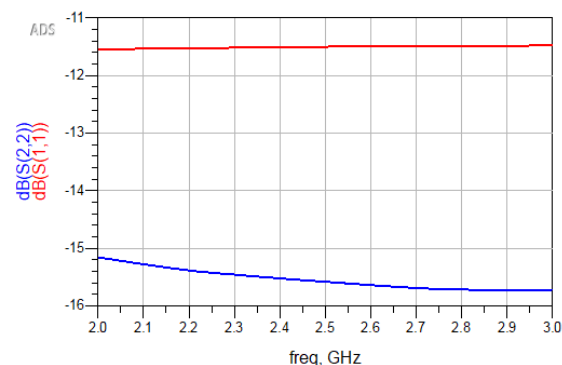
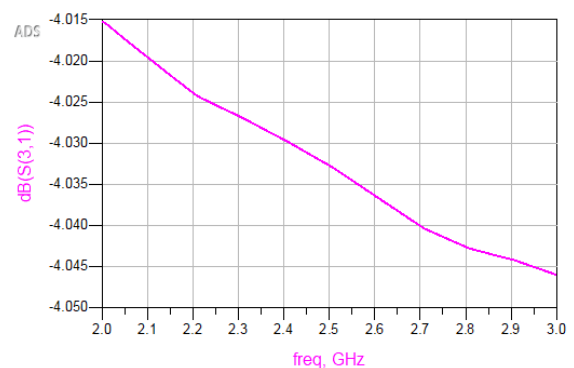
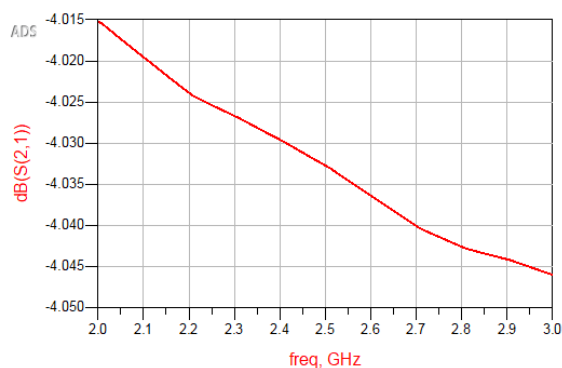
По результатам моделирования видно, что запирающие в закрытых каналах (между RFC и одним из входов, S21 или S31) становится близким к запирающему исходного SPST-ключа в 10 дБ. При этом развязка между выходами S32 получается значительная не хуже 26 дБ. Ожидается, что коэффициенты отражения как общего выхода RFC (S11), так и выходов RF1/RF2 (S22) уходят в диапазон 0..-1,5 дБ.



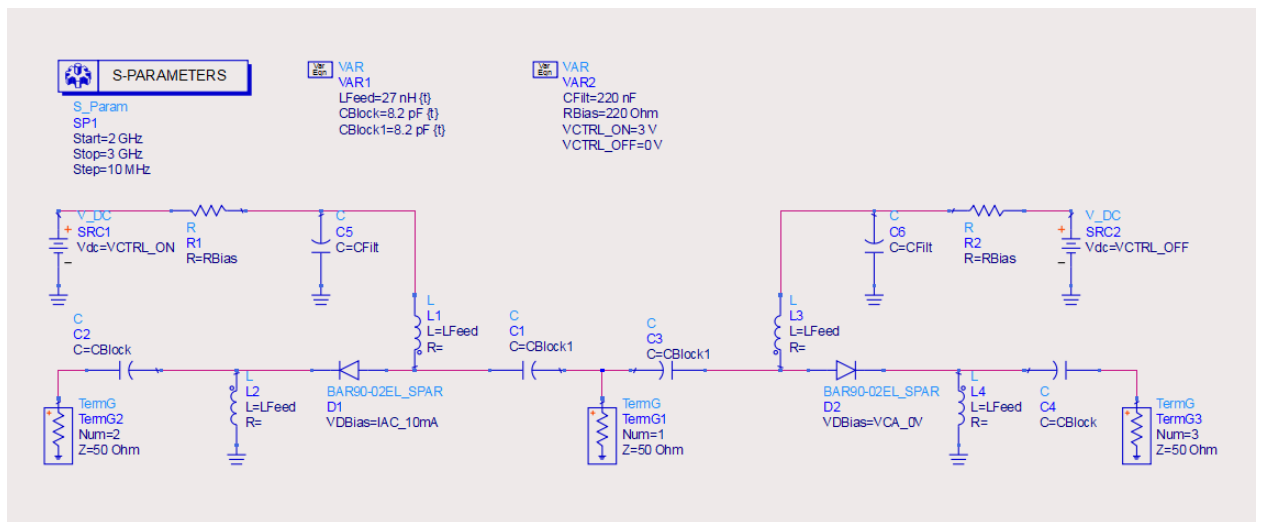
Изучим, как будет вести себя SPDT-ключ, если открыть оба канала. Данный режим как правило является для SPDT-ключей нештатным.



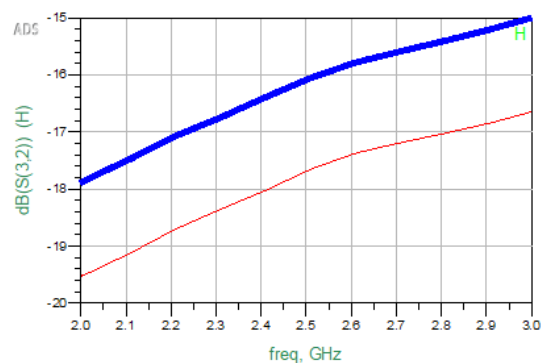
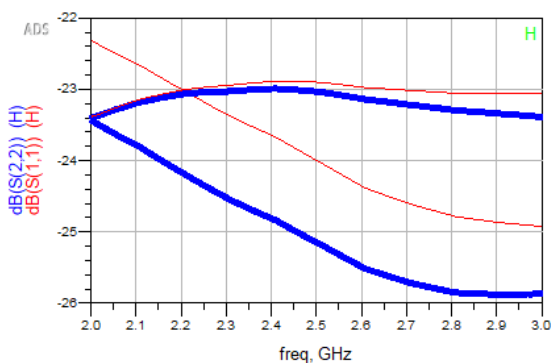
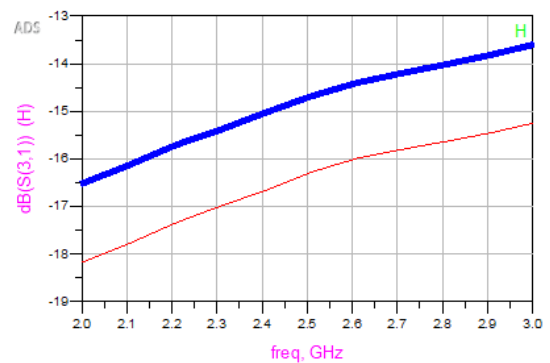
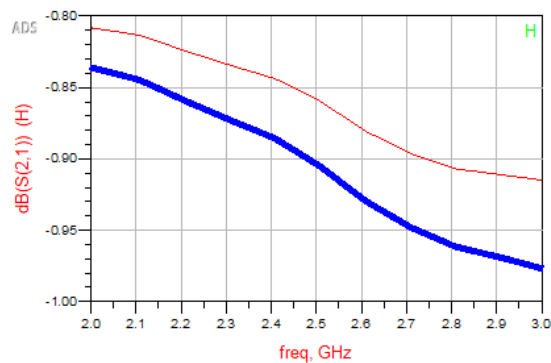
По результатам моделирования можно сказать, что SPDT-ключ ведет себя как делитель мощности с малой развязкой. Потери в открытых каналах по отношению к общему входу RFC (S21 и S31) становятся близки к 4 дБ. Развязка между выходами S32 получается малая порядка 5,4 дБ. При этом все выходы (S11, S22 и S33) можно считать согласованными, S11 порядка – 11,5 дБ, S22 и S33 порядка –15 дБ.



Также можно изучить, как изменяются характеристики SPDT-ключа в основном режиме, если чуть упростить схему управления и в качестве запирающего напряжения брать 0 В, а не –3 В.



Для быстрой оценки можно воспользоваться режимом истории в области графиков по отношению к обычному режиму работы.



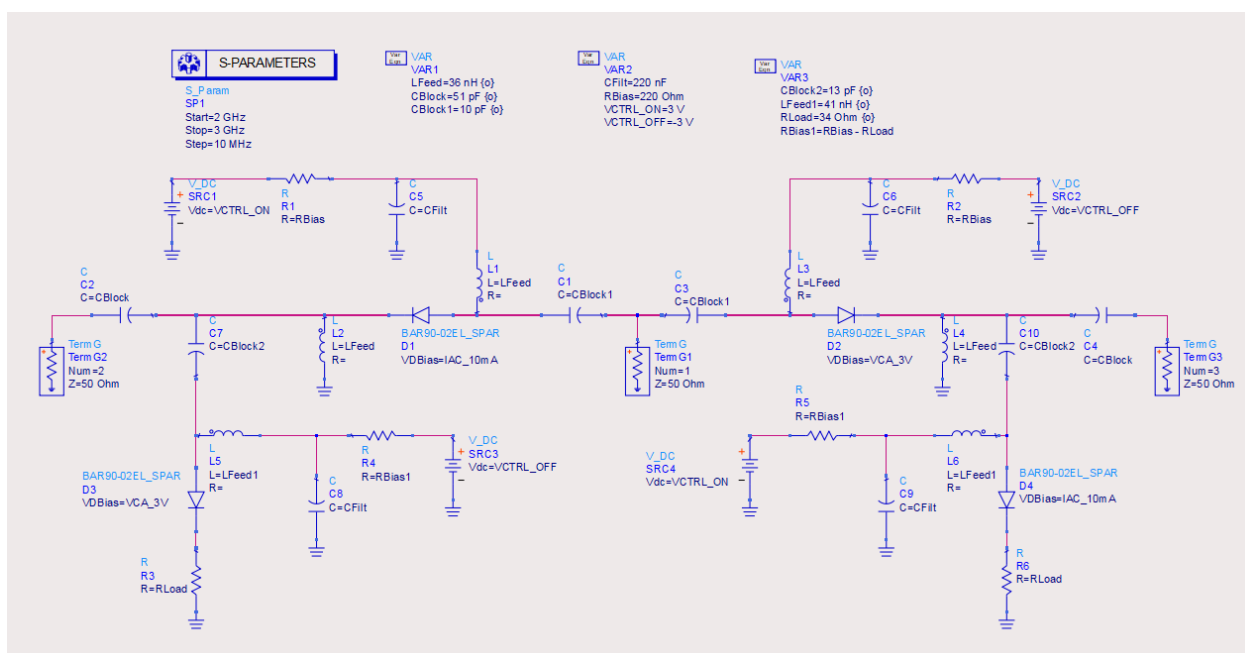
По результатам моделирования видно, что незначительно увеличиваются потери в открытом канале S21, практически не изменяется согласование на открытом входе S22 при небольшом улучшении согласования на общем S11. При этом почти на 2 дБ ухудшаются записания, как для закрытого канала S31, так и развязка между выходами S32, что уже значительно. Поэтому все-таки остановимся на схеме управления  $VCTRL = +3 \text{ В}/-3 \text{ В}$ .

## Проектирование неотражающего SPDT-ключа

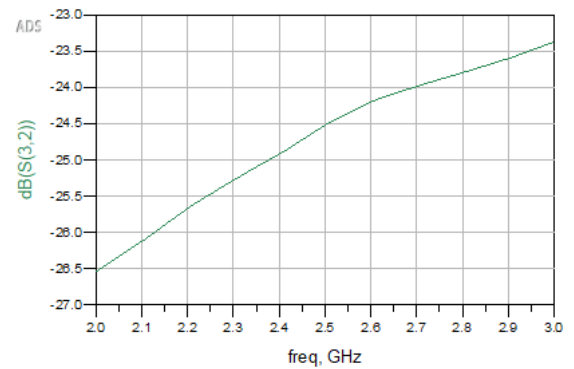
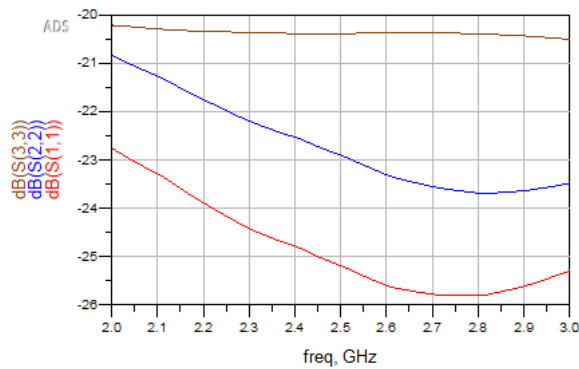
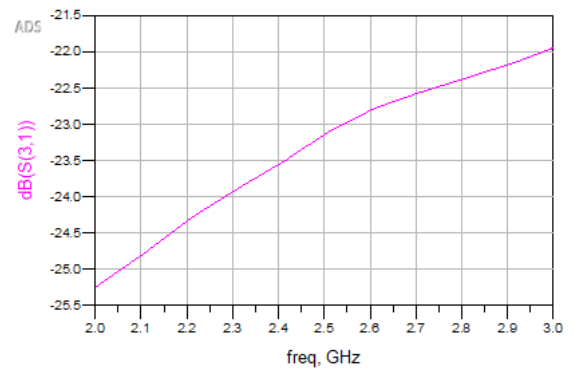
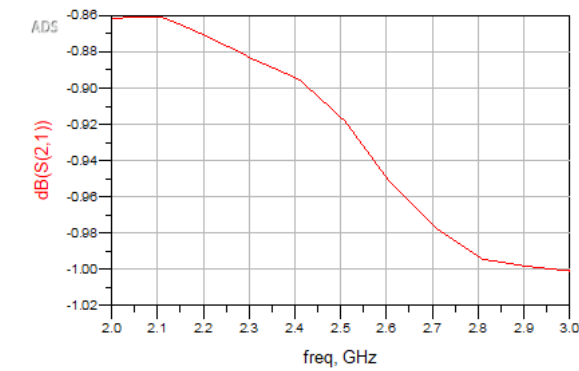
В некоторых применениях ключей недопустимо, чтобы коэффициент отражения разомкнутого канала был 0 дБ, т.е. разомкнутый канал должен быть неотражающим.

Создадим схему SW\_SPDT\_NoRef1, в которой построим неотражающий ключ на основе спроектированного ранее SPDT-ключа. Для этого добавим еще одно парное управляемое звено, замыкающее закрытый канал в дополнительную нагрузку. Это звено будет включаться в противоположное состояние к своему каналу. За первую прикидку также берем номиналы компонентов из SPST-ключа. Но т.к. скорее всего потребуется отдельное согласование, то для этого звена номинал блокировочного конденсатора вынесен в отдельную переменную CBlock2, номинал индуктивности в переменную LFeed1. Также, необходимо добавить некоторую нагрузку RLoad, в которую будем отправлять сигнал с неиспользуемого входа. При этом это значение нужно вычесть из номинала токоограничивающего резистора RBias чтобы корректно задавать прямое смещение  $I_F = 10 \text{ мА}$ , т.е.  $R_{Bias1} = R_{Bias} - R_{load}$ .

На рисунке показана финальная настроенная схема. При подборе нужно следить, чтобы характеристики не испортились по отношению к SPDT-ключу, плюс нужно добиться хорошего коэффициента отражения отключенного канала S33. Некоторые номиналы возможно придется значительно изменить.

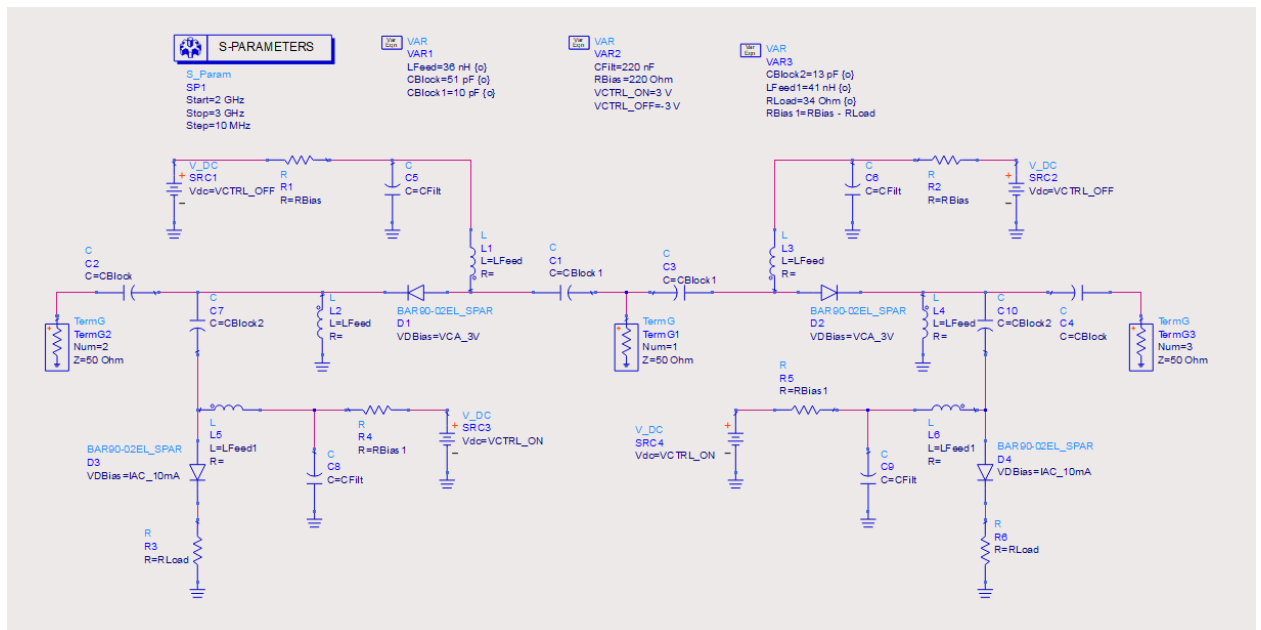


Результаты настроенной схемы показаны ниже.



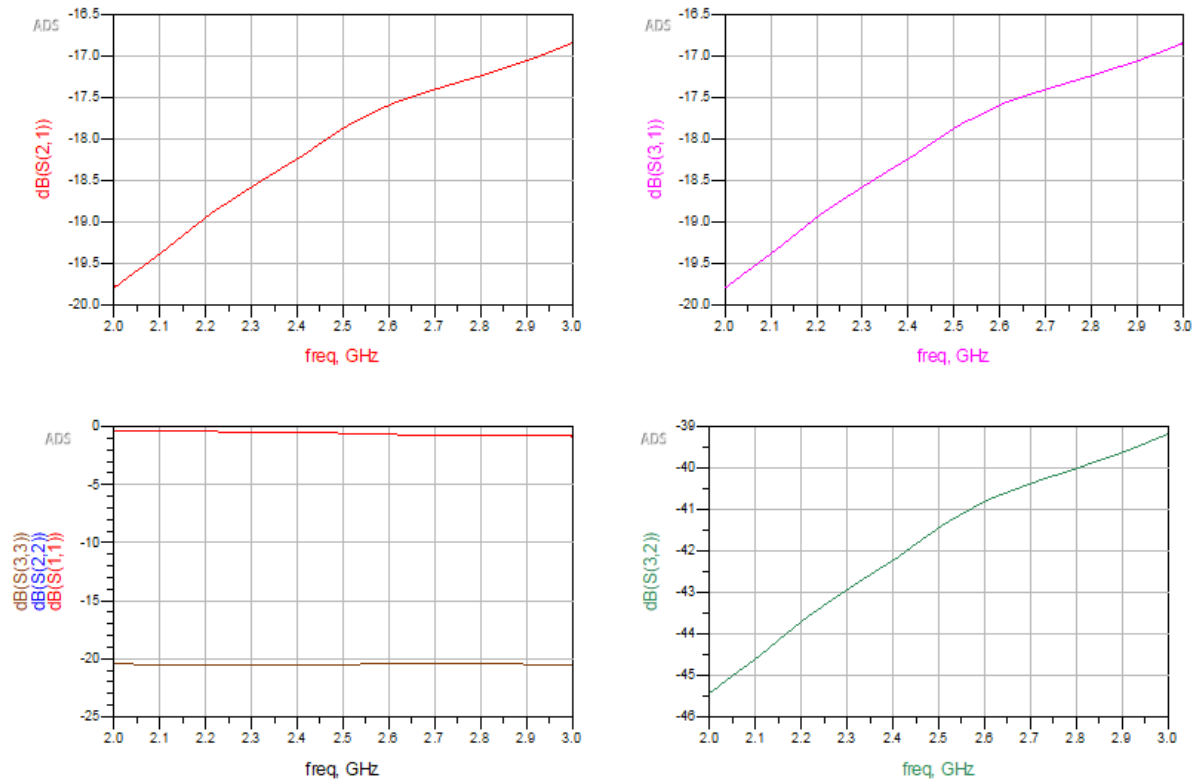
Неотражающий SPDT-ключ готов.

Теперь изучим, как работает спроектированный неотражающий SPDT-ключ, если отключить оба канала.

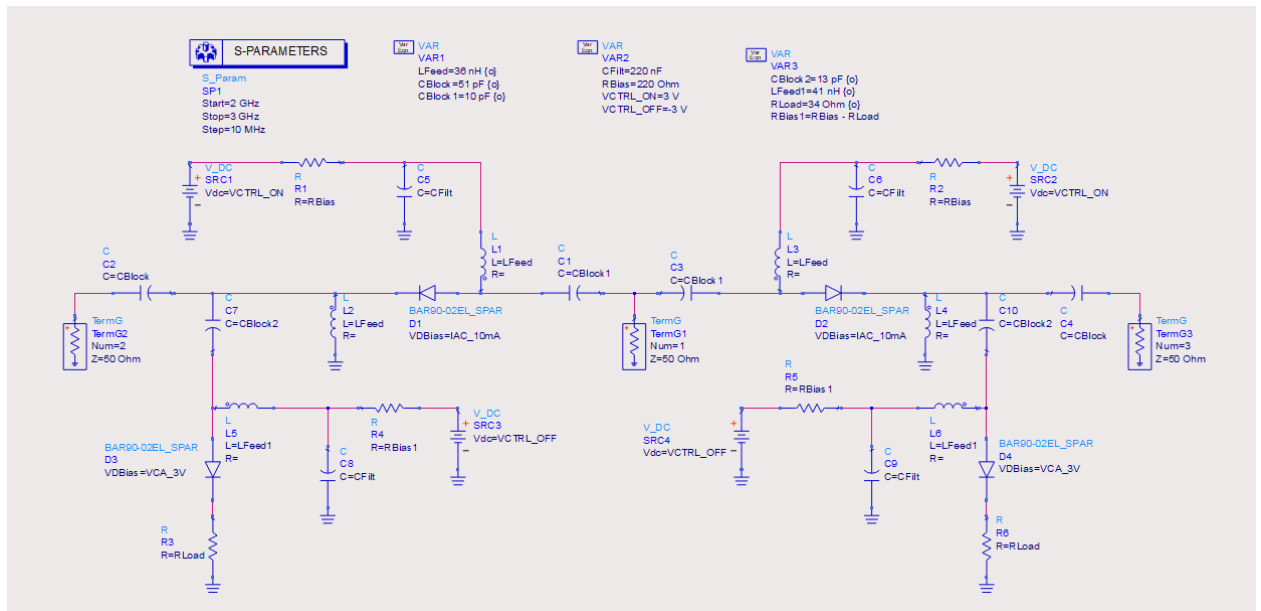


По результатам видно, что запирающие отключенных каналов (S21 и S31) ухудшаются до 16,5 дБ при громадной развязке S32 в 39 дБ. В тех выходах, где добавлено поглощающее звено (S22 и S33), коэффициент отражения лучше -20 дБ. А вот на общем выходе RFC, где нет поглощающего звена, S11

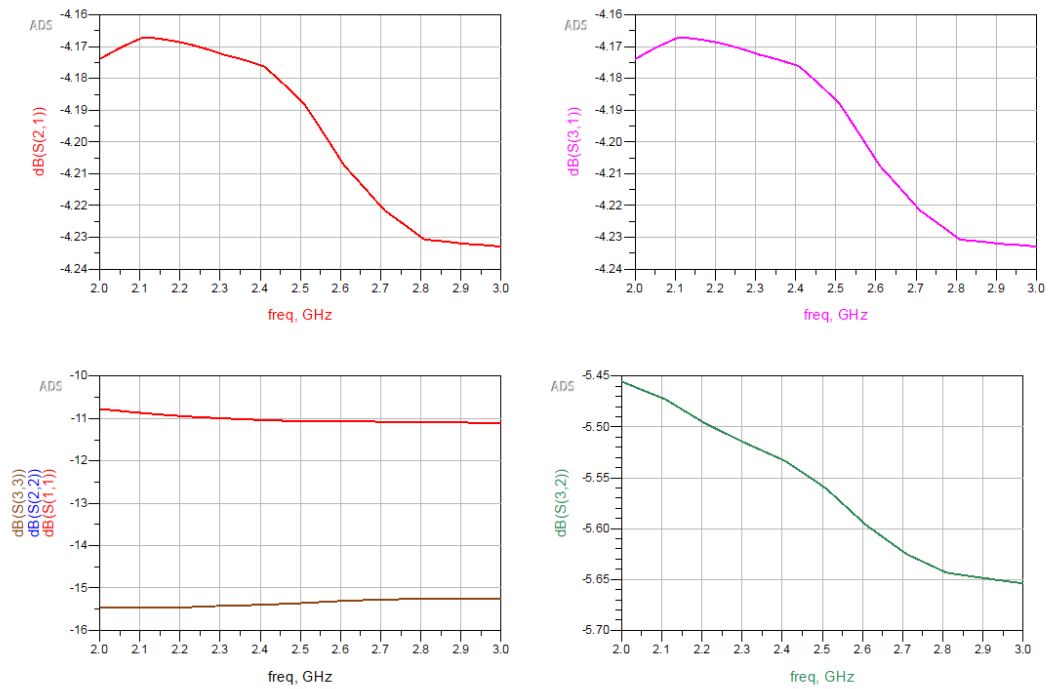
стремится к 0 дБ. Если это неприемлемо, то на входе RFC также надо будет добавить свое поглощающее звено.



Окончательно изучим, что будет, если включить оба канала.



Результаты получаются аналогично использованию обычного SPDT-ключа. Результаты похожи на неразвязанный делитель мощности.



Нужно помнить, что в проекте мы никак не учитывали влияние микрополосковых цепей между компонентами, использовали для дискретных компонентов идеальные модели и вообще все моделирование проводилось в линейном режиме. При учете этих особенностей, результаты могут оказаться несколько хуже.



### Задание на выполнение

В соответствии с вариантом спроектировать несколько видов высокочастотных ключей на PIN-диодах [6, 9]. При подборе номиналов дискретных компонентов ориентироваться на оптимум между согласованием и потерями в открытом канале, запираем закрытого канала и развязкой.

Варианты заданий приведены в таблице 3.

При выполнении ориентироваться на методику выполнения.

**Таблица 3. Варианты заданий**

№ Варианта	PIN-диод	Частотный диапазон
1	BAR67-02V	3,0..3,4 ГГц
2	BAR50-02V	2,0..2,4 ГГц
3	BAR65-02V	1,7..2,0 ГГц
4	BAR63-03W	2,1..2,4 ГГц
5	BAR63-02V	2,3..2,6 ГГц
6	BAR64-02EL	2,6..3,0 ГГц
7	BAR64-02V	2,2..2,7 ГГц
8	BAR65-03W	1,2..1,7 ГГц
9	BAR63-02L	2,3..2,6 ГГц
10	BAR64-03W	2,0..2,4 ГГц

Каждый крупный этап рекомендовано делать в отдельных ячейках (схемах), т.к. при выполнении работы возможно придется возвращаться к предыдущим этапам. Рекомендовано использовать использованное в методике именование схем.

### **Этапы выполнения:**

1. Создание проекта и подготовка компонента-хранилища S-параметров PIN-диода:

- создать проект со стандартной технологией;
- в основной библиотеке проекта `_lib` создать подпапку `circuit\data` (или `circuit\models`), прописать ее в настройках библиотеки проекта в разделе `Input data directories` и распаковать в эту папку S-параметры используемого PIN-диода [9];
- создать ячейку компонента-хранилища, формат именования ячейки «%DiodeName\_SPAR», где `%DiodeName` – краткое обозначение используемого PIN-диода;
- исходя из именования файлов S-параметров определиться с шаблоном сборки имени подключаемого файла с помощью функции `strcat()`;
- в ячейке `%DiodeName_SPAR` создать блок `S1P` и два пина, для анода и катода. Вывод 1 блока `S1P` подключать к пину-аноду;
- в ячейке `%DiodeName_SPAR` создать символ диода;
- перейти в настройки ячейки `%DiodeName_SPAR`, изменить позобозначение по умолчанию на `D`, создать список текстовых параметров `VDBias`, `PartNumer`, `Configuration`, `SPARFreqRange` и `Package`;
- параметры `PartNumer`, `Configuration` и `Package` заполнить на основании документации используемого PIN-диода;
- параметр `SPARFreqRange` заполнить на основании частотной сетки в S-параметрах используемого PIN-диода;
- по кнопке `Save AEL file` создать в папке с ячейкой текстовый файл `itemdef.ael`, описывающий ячейку `%DiodeName_SPAR` и ее параметры;
- отредактировать файл `itemdef.ael`, в котором до вызова функции `create_item()` создать список именованных констант, соответствующих прямым и обратным смещениям, при которых измерены S-параметры;
- объединить все созданные именованные константы в перечисление;
- отредактировать аргументы функции `create_item()`, определяющие параметры с помощью функции `create_parm()`;

- для параметра VDBias установить битовую маску 0, выбор из созданного ранее перечисления, тип единицы безразмерную со значением по умолчанию первой из именованных констант;

- параметрам PartNumber, Configuration, SPARFreqRange и Package установить битовую маску 577 (или [PARM\\_NOT\\_EDITED](#) | [PARM\\_NOT\\_NETLISTED](#) | [PARM\\_NO\\_DISPLAY](#)).

## 2. Выбор рабочих смещений (схема SW\_ChooseBias):

- собрать схему для выбора прямого и обратного смещений PIN-диода;
- анализировать поведение во всем проектируемом частотном диапазоне;

- прямое смещение IF выбирать исходя из минимального коэффициента отражения. При прочих равных выбирать меньшее по прямому току значение;

- обратное смещение выбирать из списка часто встречающихся управляющих напряжений 3 В и 5 В, а также 0 В по критерию лучшего записывания коэффициента отражения ближе к 1. При прочих равных выбирать меньшее по номиналу обратное смещение VR;

- если обратное смещения VR выбрано 0 В, то положительное напряжение управления выбирать 3 В или 5 В. И тогда управляющее напряжение будет  $VCTRL = +3\text{ В (или } 5\text{ В)}/0\text{ В}$ ;

- если обратное смещение выбрано ненулевое, то определить номиналы управляющих напряжений симметричными вокруг 0, т.е.  $VCTRL = \pm VR$ .

## 3. Расчет токоограничивающего резистора:

- на основании документации производителя определить, какому прямому падению напряжения VF соответствует выбранный ток прямого смещения IF;

- исходя из выбранного положительного управляющего напряжения VCTRL\_ON и выбранного режима прямого смещения PIN-диода, рассчитать номинал токоограничивающего резистора VBias для открытого режима. Посчитанный номинал резистора округлить до ряда E24, E48 или E96.

## 4. Проектирование SPST-ключа (схема SW\_SPST):

- собрать схему SPST-ключа;

- режим моделирования установить S-параметров с частотным диапазоном проектирования;
- схему собрать двойной, верхний канал будет соответствовать замкнутому режиму (порты 1 и 2, PIN-диод открыт), нижний – закрытому (порты 3 и 4, PIN-диод закрыт);
- подобрать номиналы LC-цепи LFeed-CBlock, при расчете брать их симметричными, номиналы брать из ряда E24 или с малым шагом при сближении к единицам пФ или нГн;
- при подборе контролировать степень согласования S11 (не хуже - 10 дБ) и потери S21 (не хуже 1,5..2 дБ) открытого канала и запирающее в закрытом канале S43 (от 10 дБ и лучше);
- модифицировать схему SW\_SPST до полной, добавив цепь подачи смещения LFeed – CFilt – RBias – VCTRL;
- номинал RBias брать ранее рассчитанный, CFilt порядка 150..330 нФ;
- при необходимости подстроить номиналы компонентов, чтобы достичь целевых характеристик SPST-ключа.

#### 5. Проектирование SPDT-ключа (схема SW\_SPDT):

- с использованием схемы SW\_SPST как одиночного звена собрать SPDT-ключ;
- режим моделирования установить S-параметров с частотным диапазоном проектирования;
- для имитации обычного режима работы SPDT-ключа, один канал сделать открытым (S21), второй закрытым (S32) через выбор соответствующего смещения в PIN-диодах;
- подстроить номиналы LC-компонентов, номинал блокировочных конденсаторов со стороны общего входа RFC CBlock1 возможно придется подбирать отличающимися от выходных блокировочных конденсаторов CBlock;
- при подборе номиналов LC-компонентов контролировать потери в открытом канале S21, коэффициенты отражения общего S11 и открытого входов S22, запирающее в закрытый канал S31 и развязку между входами RF1 и RF2 S32;
- стремиться, чтобы характеристики стали не хуже, чем у одиночного звена (SPST-ключа);

– после настройки провести исследование спроектированного SPDT-ключа, в том числе:

- определить коэффициент отражения закрытого канала S33;
- как себя будет вести SPDT-ключ, если оба канала закрыты;
- как себя будет вести SPDT-ключ, если оба канала открыты;
- как себя будет вести SPDT-ключ, если изменить в закрытом канале обратное смещение.

6. Проектирование неотражающего SPDT-ключа (схема SW\_SPDT\_NoRefl):

– с использованием схемы SW\_SPDT собрать схему неотражающего SPDT-ключа за счет добавления еще одного парного звена, замыкающего отключенный канал в нагрузку;

– за основу для дополнительного звена брать схему SPST-ключа;

– режим моделирования установить S-параметров с частотным диапазоном проектирования;

– для имитации обычного режима работы SPDT-ключа, один канал сделать открытым (S21), второй закрытым (S32) через выбор соответствующего смещения в PIN-диодах;

– учесть, что в формировании прямого смещения в дополнительном звене участвуют два резистора, RLoad и RBias1, учесть это при задании их номиналов;

– при подборе номиналов RLC-компонентов контролировать потери в открытом канале S21, коэффициенты отражения всех выходов (S11, S22 и S33), запирающие в закрытый канал S31 и развязку между входами RF1 и RF2 S32;

– стремиться, чтобы характеристики стали не хуже, чем у исходного SPDT-ключа;

– после настройки провести исследование спроектированного неотражающего SPDT-ключа, в том числе:

- как себя будет вести SPDT-ключ, если оба канала закрыты;
- как себя будет вести SPDT-ключ, если оба канала открыты.

7. Проект текущей лабораторной работы будет использован в последующих работах. Необходимо сохранить проект.

## Требования к отчёту

Отчет о выполненной лабораторной работе должен представлять из себя структурированное описание всей выполненной работы, в том числе должен содержать:

- Титульный лист, с наименованием работы, указанием исполнителя и даты выполнения.
- Цель (-и) работы.
- Список использованных инструментов в лабораторной работе.
- Разделы работы, которые рекомендовано формировать в соответствии с этапами выполнения (аналогично методике и заданию на выполнение).
- Для каждого этапа необходимо привести все выполненные расчеты, подготовленные модели, результаты и анализ результатов.

Подробность описания этапов выполненной работе студент определяет самостоятельно. Самое простое правило, которому надо следовать – человек, незнакомый с выполненной работой (но знакомый с использованным инструментом) должен быть в состоянии по отчету повторить данную работу и понять, что в ней происходит.

Готовый отчет необходимо экспортировать в pdf.

Архив проекта ADS лучше всего делать встроенным инструментом File – Archive Workspace из основного окна ADS. При архивации проекта можно выбрать, какие составляющие проекта добавлять в архив. Если какие-то ячейки или результаты расчета не нужны, то их можно исключить из архивирования.

По окончании выполнения лабораторной работы и подготовки отчета, отчет и архив проекта надо выложить в ОРИОКС в домашнее задание в дисциплину, привязав к контрольному мероприятию ЗЛР (Защита лабораторных работ). Именование отчета и архива проекта должно давать возможность точно понять, к какой лабораторной работы они относятся (Например, PPU\_Lab8\_Switch вместо непонятного Lab8 или Workspace8).

### **Задание на самостоятельную работу**

#### **1) Подготовка к лабораторному занятию**

При подготовке к выполнению лабораторной работы необходимо продумать шаблон отчета, при необходимости внести краткие теоретические сведения, продумать и наметить количество, вид и расположение таблиц и графиков с измеренными данными. Для получения допуска необходимо подробно изучить теоретический материал.

2) С использованием навыков, полученных в лабораторной работе, выполнить соответствующий этап БДЗ.

## Контрольные вопросы

1. Что такое SPST-ключ? Как он работает?
2. Что такое SPDT-ключ? Как он работает?
3. Поясните назначение RCL-компонентов в SPST-ключе. Как проходит управляющий ток? Как проходит ВЧ-сигнал?
4. Как определяются рабочие смещения для PIN-диода в работе, по каким критериям?
5. Поясните расчет токоограничивающего резистора для прямого режима PIN-диода. Почему рассчитанным резистором можно пренебречь для оценки режима обратного смещения?
6. В работе для расчетов используются S-параметры PIN-диодов. К каким ограничениям при моделировании это приводит? Какие из целевых характеристик спроектированных ключей невозможно в этом случае промоделировать и почему?
7. PIN-диод в ключевом устройстве может быть поставлен последовательно, параллельно или комбинированно последовательно-параллельно. Как работают PIN-диоды в этих случаях?
8. Как можно сделать отключённый канал ключа неотражающим? Как для этого надо модифицировать схему? Покажите на примере SPST-ключа.
9. Как из измеренных S-параметров SPDT-ключа определить целевые характеристики ключа?
10. Какие будут характеристики у SPDT-ключа в режимах «оба канала включены» и «оба канала выключены»? А у неотражающего SPDT-ключа в этих же режимах?



## Литература

1. Банков, С. Е. Электродинамика для пользователей САПР СВЧ : учебник / С. Е. Банков, А. А. Курушин. — Москва : СОЛОН-Пресс, 2017. — 316 с. — ISBN 978-5-91359-236-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/107661> (дата обращения: 02.02.2021). — Режим доступа: для авториз. пользователей.

### *Перечень ресурсов сети «Интернет»*

2. Сборник примеров работы в ADS «ADS Example Book: Focused on RF and Microwave Design», доступен после свободной регистрации <https://www.keysight.com/main/editorial.jsp?cc=RU&lc=rus&ckey=2704333&id=2704333&cmpid=zzfindeesof-ads-rfmw-examples>

3. База знаний Образовательного центра Keysight EEsof EDA Knowledge Center, доступен после свободной регистрации, <http://edadocs.software.keysight.com/display/support/Knowledge+Center>

4. Тематический раздел «Rf & Microwave Design» форума electronix.ru, доступен после свободной регистрации, <https://electronix.ru/forum/index.php?showforum=63>

5. Интернет-энциклопедия разработчиков СВЧ-аппаратуры «Microwaves101» <https://www.microwaves101.com>

6. Раздел PIN-диодов Infineon <https://www.infineon.com/cms/en/product/rf/rf-diode/rf-pin-diode/>

7. Страница PIN-диола BAR90-02EL Infineon <https://www.infineon.com/cms/en/product/rf/rf-diode/rf-pin-diode/antenna-switch/bar90-02el/>

8. Документация на PIN-диод BAR90-02EL [https://www.infineon.com/dgdl/Infineon-BAR90-02EL-DS-v01\\_00-EN.pdf?fileId=5546d46265f064ff01663895cd694e6e](https://www.infineon.com/dgdl/Infineon-BAR90-02EL-DS-v01_00-EN.pdf?fileId=5546d46265f064ff01663895cd694e6e)

9. Архив S-параметров PIN-диодов Infineon <https://www.infineon.com/cms/en/product/rf/rf-diode/rf-pin-diode/antenna-switch/bar90-02el/#!?fileId=5546d46269e1c0190169ecdea94b46e8>

10. Сборник примеров проектирования СВЧ-ключей на основе PIN-диодов Infineon [https://www.infineon.com/dgdl/Infineon-AN\\_1809\\_PL32\\_1810\\_172154\\_PIN%20diodes%20in%20RF%20sw%20applicati ons-AN-v01\\_00-EN.pdf?fileId=5546d46265f064ff016643e2bc241042](https://www.infineon.com/dgdl/Infineon-AN_1809_PL32_1810_172154_PIN%20diodes%20in%20RF%20sw%20applicati ons-AN-v01_00-EN.pdf?fileId=5546d46265f064ff016643e2bc241042)

11. Репозиторий с методическими указаниями по лабораторным работам <https://github.com/dee3mon/Keysight-ADS-RFDevices-methodic/>

12. Страница IDE VSCode <https://code.visualstudio.com/>

13. Расширение AEL для VSCode  
<https://marketplace.visualstudio.com/items?itemName=keysight-technologies.ael>

***Каналы Youtube с видеоуроками по Keysight Advanced Design System***

14. Канал youtube образовательного центра Keysight EEsof EDA  
<https://www.youtube.com/user/KeysightEESOF>

15. Канал youtube Anurag Bhargava образовательного центра  
<https://www.youtube.com/user/BhargavaAnurag>

16. Канал youtube Keysight EEsof EDA Field  
<https://www.youtube.com/c/EEsofAETips>

**Разработчик:**

Ст. преподаватель Института МПСУ

Приходько Д.В.