**Лабораторная работа № 1** (редакция 09.2016, составитель А.Н. Богаченков, ©МИРЭА, каф. РСПИ) **Основы компьютерного проектирования РЭС**

**СХЕМОТЕХНИЧЕСКОЕ МОДЕЛИРОВАНИЕ РЭС**

##### Общие сведения

В лабораторной работе используется программная среда Multisim (Electronics Workbench). Особенностью программы яв- ляется наличие контрольно-измерительных приборов, прибли- женных к реальным аналогам — мультиметра, генераторов сиг- налов и кодов, осциллографа, измерителя частотных характери- стик, логического анализатора и др. Имеется встроенная обшир- ная библиотека аналоговых и цифровых электронных компонен- тов, большой набор методов анализа различных характеристик электронных схем. Muiltisim является частью пакета Circuit Design Suite, в который также входит среда Ultiboard для проек- тирования печатных плат.

##### Порядок выполнения

###### Знакомство с программной средой

**Внимание!** При закрытии схем и выходе из программы на запрос о сохранении изменений (Do you want to save …) всегда выбирать ответ «**No**».

Перед началом работы выполнить процедуру «**Инициализа- ция Multisim**», используя соответствующий значок на рабочем столе. Запустить программу посредством значка  (Multisim ...). Ознакомиться с основными элементами интерфейса программы: пунктами меню, линейками контрольно-измерительных приборов и библиотек компонентов.

|  |  |  |  |
| --- | --- | --- | --- |
| *Основные команды* | *Пункты меню* | *Значок* | *Клавиша* |
| Открытие файла схемы | File / Open |  | Ctrl + O |
| Запуск моделирования | Simulate / Run |  | F5 |
| Остановка моделирования | Simulate / Stop |  |  |
| Временная остановка | Simulate / Pause |  | F6 |

Нажав кнопку открытия файла, войти в каталог

##### D: \ Electronics Workbench \ CompDesign.

Загрузить файл **Demo - DDS.ms14** . Двойным щелчком рас- крыть приборы — Logic Analyzer (логический анализатор), Oscil- loscope (осциллограф). Запустить моделирование, освоить основ- ные элементы настройки осциллографа:

**Reverse** — переключение цвета фона окна;

**Time base** [s/Div, ms/Div, µs/Div, ns/Div] — развертка по горизонтали в единицах времени на деление;

**Channel A (B) Scale** [V/Div, mV/Div, µV/Div] — цена деле- ния по амплитуде в канале A (B);

**Channel A (B) Y pos (Div)** — смещение кривой по вертикали в делениях.

Интерактивные органы S0…S9 позволяют непосредственно в процессе моделирования изменять логические уровни. Опреде- лить по схеме, измерить с помощью осциллографа и **привести в отчете** следующие параметры: разрядность регистра аккумуля- тора … ; частота дискретизации … ; диапазон задания кода: …… (в 2-й системе); диапазон перестройки частоты генератора: …… .

Открыть проект **QuizShowProject \ quizshowproject.mp14**,

просмотреть страницы схем, модель печатного узла и др. Закрыть все открытые окна, перезапустить Multisim.

###### Освоение процесса подготовки схем

Используя встроенные библиотеки компонентов и приборов, подготовить схему по указанию преподавателя. Рекомендуется сначала разместить все компоненты на рабочем поле, выбрав их из библиотеки, вызываемой по щелчку на соответствующих значках инструментальной линейки. Затем сформировать соеди- нения — навести мышью на вывод компонента до появления точки, щелкнуть левой кнопкой и тащить до вывода другого ком- понента или проводника (можно щелкать по пути, обозначая уг- лы), щелкнуть в конечной точке. Поворот, копирование, измене- ние различных параметров производится выбором пунктов из контекстного меню компонента или цепи.

Запустить процесс моделирования, продемонстрировать пре- подавателю. **Привести в отчете** название устройства, скриншоты схемы и полученных диаграмм.

###### Снятие передаточной характеристики полевого транзистора

Передаточная характеристика представляет собой зависи- мость тока стока IС от напряжения на затворе UЗИ. Ток стока пропорционален падению напряжения на измерительном рези- сторе 1 Ом. Это напряжение подводится к входу A осциллографа. Напряжение на затворе формируется источником напряжения переменного тока и поступает на вход канала B осциллографа.

Загрузить схему 3. Запустить моделирование. **Привести в отчете** временные диаграммы IС и UЗИ .



Установить усиление по каналу B равным 2 вольт/дел и пе- реключить осциллограф из режима Y/T в режим A/B. В этом ре- жиме сигнал IС развертывает луч по оси Y (канал A), а сигнал UЗИ

* по оси X (канал B). **Привести в отчете** зависимость IС (UЗИ), нанести на оси графика деления шкалы и соответствующие им реальные значения токов и напряжений.

По графику определить величину напряжения отсечки UОТС (порог начала протекания тока стока) и максимальную крутизну характеристики S = dIC/dUЗИ или S = (IC2 – IC1) / (UЗИ2 – UЗИ1).

###### Моделирование транзисторного усилительного каскада



* 1. Загрузить схему 4. Рабочая точка транзистора, обеспечи- вающая максимальную амплитуду неискаженного выходного сигнала, задается напряжением смещения, снимаемым с потен- циометра R1 (регулировка — клавишами Z и Shift+Z). Наблюдая осциллограммы сигналов, установить оптимальную рабочую точку и не изменять ее во всех последующих экспериментах.

Ознакомиться с некоторыми встроенными в программу ме- тодами анализа, выбор которых осуществляется в диалоге, вызы- ваемом из меню **Simulate / Analysis and simulation.** Все резуль- таты в виде таблиц и графиков **привести в отчете**.

* 1. Выбрать метод **DC Operating Point** — расчет режимов по постоянному току. Задать для анализа переменные, соответ- ствующие потенциалам узлов и выходному току.
	2. Выбрать метод **AC Sweep** — расчет частотных характе- ристик (АЧХ, ФЧХ). Оставить значения по умолчанию для пара- метров Start frequency, Stop frequency, Sweep type, Number of points, Vertical scale — начальная и конечная частоты, закон из- менения частоты, число рассчитываемых точек, масштаб по вер- тикали. На вкладке Output задать для анализа узел, соответству- ющий стоку транзистора.
	3. Выбрать метод **Parameter Sweep** — моделирование при вариации заданных параметров. Задать Sweep parameter: Device parameter, Device type: Vsource, Name: V1, Parameter: dc, начальное (Start) и конечное (Stop) значения — 0 и 12 В, шаг изменения (Increment): 2 В, линейный закон изменения (Sweep variation type: Linear), метод анализа — переходные процессы (Transient). В настройках анализа (кнопка Edit analysis) проверить параметры начала и окончания (Start time, End time) — 0 и 1...2 мс. На вкладке Output задать для анализа узел, соответствующий стоку транзистора.
	4. Выбрать метод **Monte Carlo** — статистические испыта- ния по методу Монте-Карло. Проверить наличие в списке Toler- ance list компонентов, для которых заданы отклонения номина- лов. Добавить (если отсутствует) источник питания V2, напри- мер, с 5% отклонением. На вкладке Analysis parameters задать режим анализа на постоянном токе (DC Operating Point), число испытаний (Number of runs) не менее 10, выходной узел – сток.
	5. Измерить коэффициент усиления устройства. Подклю- чить параллельно резистору в цепи истока конденсатор емкостью не менее 1 мкФ. Изменением уровня входного напряжения до- биться неискаженной передачи сигнала, снова измерить коэффи- циент усиления. Объяснить результат.

###### Моделирование работы автогенератора.



* 1. Загрузить схему 5. Зарисовать примерный вид осцилло- грамм и по возможности точнее определить частоты генерации для двух типов транзисторов — 2N2222 и 2N914. Для измерения периода сигнала рекомендуется использовать визирные линии. Замену транзисторов удобно производить, выбрав пункт кон- текстного меню компонента Replace components. Выписать ха- рактеристики транзисторов (в окне свойств транзистора на вкладке Value нажать кнопку Edit model). Заполнить и **привести в отчете** нижеприведенную таблицу. Теоретическое значение частоты генерации (при использовании идеального транзистора):

f = 1 / [2· (L·C)1/2], где C = C1·C2 / (C1+C2).

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Транзистор2N2222 | Транзистор2N914 | Идеальный транзистор |
| Измеренный период колебаний, нс |  |  | **—** |
| Частота колебаний, МГц |  |  | рассчитать |
| Емкость перехода база-эмиттер при нулевом напряжении (CJE), пФ |  |  | указать |
| Емкость перехода база-коллектор при нулевом напряжении (CJC), пФ |  |  | указать |

Продолжить знакомство с методами анализа, выбрав любой из транзисторов.

* 1. Выбрать метод **Transient** — расчет переходных процес- сов. Задать диапазон времен (Start time … End time) 0…30 мкс, максимальный шаг по времени (Maximum time step) не более 10 нс, выходной узел. **Привести в отчете** диаграмму сигнала.
	2. Выбрать метод **Temperature Sweep** — температурные испытания. Задать диапазон 0…100°C с шагом 10°C, метод ана- лиза DC Operating Point. **Привести в отчете** зависимость напря- жения на эмиттере транзистора от температуры.

##### Содержание отчета

Схемы исследуемых устройств, диаграммы, таблицы (пере- числены в каждом пункте выполнения работы). Диаграммы должны иметь шкалы, обозначения величин и единиц измерений по осям, а при наличии нескольких кривых — обозначения (по- яснения) для каждой кривой.

##### Библиографический список

1. Шестеркин, А.Н. Система моделирования и исследования радиоэлектронных устройств Multisim 10 — М.: ДМК Пресс, 2012. — 360 с.
2. Введение в Multisim. Трехчасовой курс [Электронный ре- сурс]. // National Instruments, 2007. — 42 с. — Режим доступа: ftp://ftp.ni.com/pub/branches/russia/software/multisim\_gettingstarted.pdf.
3. Схемотехническое моделирование [Электронный ресурс].

// Радиоежегодник, 2013, вып. 23. — Режим доступа: [http://www.rlocman.ru/radioyearbook/.](http://www.rlocman.ru/radioyearbook/)

##### Контрольные вопросы

###### Общие вопросы

* Назначение и возможности программы.
* Основные элементы интерфейса программной среды.
* Перечислите радиокомпоненты, содержащиеся во встроенных библиотеках.
* Дайте характеристику используемой измерительной аппаратуры. В чем отличие режимов Y/T и A/B осциллографа?
* Как осуществляется динамическое управление параметрами ком- понентов (переменными резисторами, переключателями, регули- руемыми источниками)?

###### Подготовка схем

* Как разместить и ориентировать компонент на рабочем поле? Как выполнить межкомпонентные соединения?
* Как задать параметры компонента?
* Перечислите основные и дополнительные параметры, которые могут задаваться в программе для следующих компонентов: рези- стора, конденсатора, индуктивного элемента, источника тока и напряжения.
* Как изменить параметры активных компонентов (транзисторов,

микросхем)?

###### Методы моделирования

* Что определяется в результате расчета режимов по постоянному току? Чем заменяются при этом такие компоненты как конденса- торы, индуктивности, источники сигналов?
* Как производится расчет амплитудно-частотных и фазочастотных характеристик?
* Каковы исходные данные для расчета переходных процессов?
* С какой целью производится моделирование при вариации пара- метров схемы? Перечислите возможные вариационные параметры для пассивных и активных компонентов? Объяснить получаемый результат на примере конкретной схемы.
* Какова цель испытаний по методу Монте-Карло и других стати- стических испытаний?
* Как проводятся температурные испытания?

###### Снятие характеристик транзистора

* Каким образом осуществляется вариация параметров при снятии передаточных характеристик?
* Как производится регистрация токов и напряжений? Какова цена деления по осям координат?
* Чем ограничена максимальная величина тока транзистора?
* Какой физический смысл имеет крутизна характеристики?

###### Моделирование транзисторного усилительного каскада

* Приведите электрическую модель полевого транзистора.
* Как правильно выбрать напряжение смещения на затворе транзи- стора?
* Определите минимально и максимально возможные мгновенные напряжения на стоке транзистора при большом уровне входного сигнала.
* Каков теоретический коэффициент усиления при отсутствии кон- денсатора в цепи истока? За счет чего меньше реальный коэффи- циент усиления?
* Как и почему влияет на коэффициент усиления и форму выходно- го сигнала шунтирование конденсатором резистора в цепи истока?

###### Моделирование работы автогенератора

* Приведите электрическую модель биполярного транзистора.
* Как максимально точно измерить частоту сигнала?
* Какие элементы определяют частоту генерации?
* Как влияют параметры модели транзистора и элементы связи его с контуром на частоту и форму генерируемого сигнала?
* Объясните зависимость напряжения в узлах схемы от температу- ры.

#### Приложение. Схема демонстрационного примера Demo - DDS

U1

A1 A2

A3 SUM\_1

A4 SUM\_2 SUM\_3

B1 SUM\_4 B2

B3 B4

C0 C4

5

3

14 4

12

S0 6

2

0

15

S1

11

0

7 9

0

S2

VCC 5.0V

3

10

13

1

1

9

Адрес

U4

2

4

1Q

2Q

3Q

4Q

5Q

6Q

1D

2D

3D

4D

5D

6D

~CLR CLK

5

6

7 10 12 15

14

13

11

#### Генератор синусоидального сигнала по методу прямого цифрового синтеза (DDS)

U6

S3 74HC283D\_6V

0

74HC174D\_6V

ROM

A0 D0

A1 D1

A2 D2

A3 D3

A4 D4

A5 D5

A6 D6

A7 D7

U2

S4

A1 A2

A3 SUM\_1

A4 SUM\_2 SUM\_3

B1 SUM\_4 B2

B3 B4

C0 C4

1

5

3

S5 4

12

14

0

1

6

13

0

S6 10

15

2

S7

7

11

1

9

ROM\_256x8

VCC 5.0V

 Vref+

Vref-

D0 D1 D2 D3 D4 D5 D6

D7

VDAC8

U7

Output

R1

1kΩ

74HC283D\_6V

U3

A1 A2

A3 SUM\_1

A4 SUM\_2 SUM\_3

B1 SUM\_4 B2

B3 B4

C0 C4

5

3

4

0

14

S8

10

13

6

1

12

S9

2

0

15

11

U5

3 2

4

1Q

2Q

3Q

4Q

5Q

6Q

1D

2D

3D

4D

5D

6D

~CLR CLK

5

6

7 10 12 15

14

13

11

1

9

XLA1

1

F

GND

XSC1

A B

+ \_ + \_

Ext Trig

+

\_

C1

1nF GND

GND

7 9

74HC283D\_6V

G1

384kHz

74HC174D\_6V

**C Q T**

GND

Код задания частоты Сумматор кодов аккумулятора и задания частоты

Генератор частоты дискретизации

Регистр - аккумулятор фазы

ПЗУ с таблицей значений синусоидальной функции

Цифро-аналоговый преобразователь

Фильтр НЧ

**Лабораторная работа № 3** (редакция 09.2015, составитель А.Н. Богаченков, ©МИРЭА, каф. РСПИ) **Основы компьютерного проектирования РЭС**

**ОСНОВНЫЕ ПРИЕМЫ ПРОЕКТИРОВАНИЯ В СРЕДЕ PROTEUS**

## Общие сведения

В лабораторной работе используется программный пакет Proteus Design Suite — система автоматизированного проектиро- вания от британской компании Labcenter Electronics Ltd. Пакет представляет собой систему схемотехнического моделирования, базирующуюся на основе моделей электронных компонентов, принятых в PSpice. Особенностью пакета является возможность моделирования смешанных цепей, анимация компонентов, симу- ляция работы программируемых устройств (микроконтроллеров, сигнальных процессоров, логических матриц), использование виртуальных приборов. Библиотека содержит несколько тысяч аналоговых и цифровых компонентов. Дополняет пакет система проектирования печатных плат. Proteus Design Suite состоит из двух основных частей, которые в последних версиях входят в единую оболочку:

ISIS — среда подготовки, синтеза и моделирования элек- тронных схем; включает схемотехнический редактор и виртуаль- ную систему моделирования (Proteus VSM);

ARES — среда разработки печатных плат.

## Порядок выполнения

### Знакомство с программной средой

**Внимание!** При загрузке новых схем и выходе из программы на запрос о сохранении изменений всегда выбирать ответ «**Нет**».

Перед началом работы выполнить процедуру «**Инициализа- ция Proteus**», используя соответствующий значок на рабочем столе. Запустить программу ISIS значком .

##### Некоторые приемы работы в среде

*Способы масштабирования*:

* роликом мыши над местом, которое необходимо увеличить или уменьшить;
* клавишами F6 (увеличение), F7 (уменьшение) при наведении мыши на нужное место;
* выделением требуемой области левой кнопкой мыши при нажатой клавише Shift;
* кнопками инструментальной панели    ;
* клавишей F8 — для отображения чертежа целиком.

*Способы панорамирования*:

* средней кнопкой мыши (или кнопкой колесика);
* установкой указателя мыши в будущий центр и нажатием F5;
* при нажатой клавише Shift — сдвигом указателя мыши к гра- ницам окна редактирования;
* щелчком левой кнопкой мыши в окне краткого обзора;
* кнопкой панели инструментов .

*Выбор и размещение компонентов на схеме*:

Нажать на кнопку левой вертикальной панели, далее на кнопку P (либо выбрать из кон- текстного меню "Разместить/Компонент…"). Появляется диалоговое окно "Pick Devices" вы-

бора элемента из библиотеки. При двойном щелчке по строке описания элемента он будет перенесен в список селектора устройств. Рекомендуется выбирать те элементы, для которых существует модель и посадочное место (корпус).

Для размещения компонента на поле чертежа выбрать его в списке, щелкнуть на схеме, позиционировать, при необходимости повернуть (клавишами "+", "-" ), еще раз щелкнуть. Действия над уже размещенным компонентом — после однократного щелчка по нему.

Создание соединений: снять выделение со всех элементов (щелкнуть по пустому месту схемы), навести курсор на вывод (курсор в виде зеленого карандаша), щелкнуть левой кнопкой, вести к нужной точке (можно щелкать по пути, обозначая углы), щелкнуть в конечной точке.

*Кнопки*: запуска моделирования, пошаговой работы, паузы и остановки моделирования: 

##### Ознакомление с демонстрационными примерами

Загрузить нижеперечисленные проекты (они и множество других примеров входят в состав программного пакета). Ознако- миться с виртуальными приборами (вольтметр, амперметр, гене- ратор сигналов, осциллограф, логический анализатор), органами интеллектуального моделирования (переключатели, регуляторы). При запуске моделирования не следует закрывать возникающие окна. Если не открывается окно виртуального прибора, щелкнуть правой кнопкой по значку прибора и выбрать его название из контекстного меню.

##### D:\Proteus\Lab1\Demo-AC.DSN

Дополнительно: поставить процесс на паузу; щелкая по ком- понентам, просмотреть их текущее состояние, **привести в отчете** максимальное значение тока через транзистор.

##### D:\Proteus\Lab1\Demo-FigLissajous.DSN

Освоиться с регулировками генераторов сигналов и осцилло- графа; **привести в отчете** 2-3 фигуры Лиссажу при кратных со- отношениях частот X/Y (X, Y = 2…6, X  Y).

##### D:\Proteus\Lab1\Demo-Clock.DSN

В проекте обратить внимание на возможность моделирова- ния символьных дисплеев, средств отображения текущих уров- ней всех логических сигналов в схеме. Настроить часы на теку- щее время. Оценить точность хода. Перенести на схему виртуальный прибор — логический анализатор, для

чего вызвать список приборов кнопкой на левой пане- ли, выбрать LOGIC ANALYZER, щелкнуть дважды на свободном месте схемы (для освобождения места мож- но удалить текстовой бокс).

Подключить 5…7 входов анализатора к точкам схемы с изменяющимися уровнями (ре- комендуется: к тактовому генератору, сигналу управления миганием двоеточия, выходам мик- росхемы U13). Шкалы анализатора поставить, как показано на рисунке. Переключатель "Cap- ture Resolution" задает период дискретизации записываемых отсчетов (он не должен превы- шать минимальную длительность регистрируе- мых импульсов). Переключатель "Display Scale" масштабирует диаграмму.

Внимание! Анализатор (в отличие от ос- циллографа) требует ручного старта. Нажатие кнопки "Capture" начинает процесс захвата сиг-

нала — цвет кнопки меняется на розовый, анализатор непрерыв- но заполняет отсчетами первую половину буфера памяти (это данные, предшествующие моменту срабатывания триггера запус- ка). Условия запуска задаются переключателями для каждого входного сигнала:

    

Всегда По уровню По фронту По уровню По спаду

"0" "0""1" "1" "1""0"

При наступлении триггерного события (одновременно по всем заданным условиям!) запись сигнала продолжается во вто- рую половину буфера (цвет кнопки — зеленый), по заполнению буфера запись останавливается, кнопка принимает исходное со- стояние, диаграмма обновляется записанными данными. В поло- жении регулятора "Position", равном 50, на дисплее анализатора видна вертикальная пунктирная линия, соответствующая усло- вию запуска. При включенной кнопке "Cursor" можно проводить измерения параметров сигналов.

##### D:\Proteus\Lab1\Demo-Motor\HC11PWM.DSN

Посредством органов управления — потенциометра и пере- ключателя проверить различные режимы работы устройства. Построение графиков переходных процессов осуществляется при остановленном моделировании, наведении указателя мыши на диаграмму и нажатии клавиши "Пробел". Во время моделирова- ния нажать клавишу "Пауза", пройти несколько шагов по про- грамме, нажимая клавишу F10.

### Методы моделирования аналоговых устройств

Для знакомства с основными методами анализа выбран ти- повой усилитель звуковой частоты из проекта

##### D:\Proteus\Lab1\SoundAmplifier.DSN

Усилитель содержит классический пассивный регулятор тембра низких и высоких частот с диапазоном регулирования на крайних частотах до ±20 дБ. Номиналы элементов верхних и нижних плеч фильтров соотносятся между собой в 10 раз. Для получения линейной АЧХ такое же отношение должны иметь сопротивления участков переменных резисторов. Для удобства измерений ослабление регулятора компенсируется таким же зна- чением коэффициента передачи усилительной части устройства.

Проверить работу устройства, наблюдая осциллограммы и переводя регуляторы из одного крайнего положения в другое (наиболее удобно осуществлять перестройку, наведя указатель мыши на потенциометр и нажимая клавиши Page Up, Page Down). Оформление осциллограмм в отчете не требуется.

##### Анализ статического режима

Определить и **привести в отчете** режимы по постоянному току активных компонентов — микросхемы и транзисторов. В состоянии паузы моделирования, щелкая по компонентам, зафик- сировать напряжения на их выводах. Для исключения влияния переменных сигналов использовать один из следующих приемов: либо обнулить уровни источников сигналов (не забыть восстано- вить их для последующей работы), либо сделать только един- ственный (первый) шаг моделирования.

Результаты проводимых далее анализов представляются на графиках-диаграммах, список графиков выводится при нажатии на значок  . После проведения очередного анализа реко- мендуется удалить диаграмму с экрана для осво-

бождения места.

##### Анализ переходных процессов

Выбрать в списке графиков "ANALOGUE" или "MIXED", на свободном поле чертежа сделать щелчок левой кнопкой мыши, движением указателя обозначить рамку и еще раз щелкнуть.

Установить на входе и выходе схемы щупы (пробники) напряжения, для чего активизировать кнопку  на левой ин- струментальной панели (другой вариант — щелкнуть правой кнопкой на свободном поле схемы и выбрать из контекстного меню пункт "Разместить / Щуп напряжения"), далее указать на нужные цепи, дать щупам осмысленные названия (хотя бы Input, Output). Поочередно выделяя щупы (однократным щелчком), перенести их на поле диаграммы (зажав левую кнопку мыши). Добавить кривую к диаграмме можно также из контекстного ме- ню диаграммы ("Добавить щуп наблюдения"). **Задать время** регистрации — определить самостоятельно исходя из размеще- ния минимум двух периодов самого низкочастотного входного сигнала, время указать в соответствующих полях диалога, вызы- ваемого из контекстного меню диаграммы ("Редактирование свойств" или "Редактировать график").

Запустить моделирование: навести указатель мыши на диа- грамму и нажать клавишу "Пробел". Дополнительные возможно- сти настройки вида диаграммы — после максимизации окна при щелчке по зеленому заголовку левой кнопкой.

Если отдельные кривые на диаграмме накладываются друг на друга, их можно сместить по вертикали. Двойной щелчок по названию кривой открывает диалог с выражением для построе- ния данной зависимости — по умолчанию это просто имя P1, к которому можно добавить константу (а в принципе написать и любую сложную математическую формулу).

**Привести в отчете** три диаграммы: при среднем и двух крайних положениях одного из регуляторов. Разнести кривые в случае их наложения. Указать значения частот входного сигнала.

##### Частотный анализ

Создать график "FREQUENCY". Перенести щуп напряжения с выхода устройства на левую часть графика (для АЧХ) и еще раз

* на правую (для ФЧХ), а также перенести на график один из источников сигнала F1, F2 или F3 (если на схеме источника нет, его надо создать). Такой источник является опорным и в процессе анализа всегда имеет единичный выходной уровень (0 дБ). Запу- стить пробное моделирование клавишей "Пробел". В свойствах диаграммы ограничить максимальную частоту значением 50…100 кГц, а также задать принудительный масштаб по оси Y для АЧХ (-20…+20 дБ).

**Привести в отчете** диаграммы для трех режимов: а) макси- мально линейной АЧХ (9-10% положение регуляторов); б) мак- симального подъема НЧ и ВЧ; в) максимального спада НЧ и ВЧ. Изобразить все диаграммы **на общих осях**.

##### Анализ шумов

Создать график "NOISE". Перенести на него щуп напряже- ния с выхода устройства, данная кривая будет отображать уро- вень выходного шума. Для оценки уровня шумов, приведенных ко входу, необходимо разместить на схеме источник переменного напряжения (SINE), подключить ко входу, дать ему осмысленное название, перенести на диаграмму. Все остальные источники при анализе игнорируются. Если на входе остался щуп напряжения, удалить его. Дополнительно в настройках можно ограничить полосу частот. Запустить анализ. Кроме графического представ- ления нормированного уровня шумов (уровень отнесен к квад- ратному корню частоты) более детальные сведения можно полу- чить из протокола моделирования — вызывается нажатием на поле  внизу экрана (или из главного меню "График / Просмот- реть лог"). **Привести в отчете** диаграмму, значения Total Output Noise, Total Input Noise, а также обозначения 3-4 компонентов, внесших наибольший вклад в уровень шумов.

**Удалить** созданный источник напряжения на входе.

##### Анализ искажений

Создать график "DISTORTION". Перенести щуп напряжения с выхода устройства на левую часть графика (для амплитудной зависимости). При желании можно щуп перенести и на правую часть (для фазы), но данная характеристика малоинформативна. Также необходимо перенести на диаграмму один из входных источников F1…F3. Если в свойствах диаграммы параметр IM равен 0, используется одночастотный гармонический анализ, результирующие зависимости представляют 2-ю и 3-ю гармони- ки. При задании IM = F2/F1 < 1 задействованы две входные часто- ты с указанным отношением, в результате моделирования произ- водится расчет интермодуляционных искажений с представлени- ем трех кривых с частотами F1+F2, F1-F2 и 2F1-F2. В данном виде анализа используются малосигнальные модели. **Привести в от- чете** результаты одного из видов анализа.

##### Анализ Фурье

Установить регуляторы в положение линейной АЧХ (9-10%). Создать график "FOURIER". Выполнить подготовку, аналогич- ную анализу переходных процессов, увеличив только время ана- лиза (до 20…100 мс). Дополнительно проверить амплитуды ис- точников (установить равными 1). Запустить моделирование и **привести в отчете** его результаты. Увеличить выходные уровни источников до 5…7. Повторить моделирование, результаты так- же **привести в отчете**.

##### Вариация параметров на постоянном токе

При анализе вычисляется постоянный (установившийся уро- вень) напряжения/тока в выбранных узлах схемы при изменении другого параметра, например, величины сопротивления, напря- жения источника, усиления транзистора, температуры и др.

Создать график "DC SWEEP". Перенести на него щуп вы- ходного напряжения. Проверить стабильность этого напряжения, например, при вариации сопротивления резистора R9. В свой- ствах графика задать пределы вариации 1/10…10 раз от номина-

ла, назвать вариационный параметр любым именем (по умолча- нию предлагается X). На схеме вместо номинального значения (не позиционного обозначения!) задать это имя. Запустить моде- лирование, **привести в отчете** результаты.

При вариации температуры используется глобальное имя TEMP, имена параметров модели можно узнать, открыв свойства диаграммы и дополнительное окно кнопкой "SPICE опции".

##### Вариация параметров на переменном токе

В данном методе рассчитывается семейство АЧХ при вариа- ции одного какого-либо параметра (как в методе вариации на постоянном токе).

Создать график "AC SWEEP". Выполнить подготовку, как при частотном анализе. Проверить, чтобы регуляторы не находи- лись в крайних положениях. Выбрать для вариации, например, один из конденсаторов C1…C4, в диалоге задать номинальное значение емкости, начальное и конечное значения емкости (в 3…10 раз отличающиеся от номинального). **Привести в отчете** результаты моделирования.

##### Проверка проекта на ошибки. Подготовка к проектиро- ванию печатной платы

Для проверки отсутствия формальных ошибок в схеме и компонентах рекомендуется просмотреть следующие протоколы: Design Explorer — вызывается из главного меню "Проект / Проводник проекта" или кнопкой  на панели инструментов. Проверяется список компонентов, наличие для всех компонентов,

устанавливаемых на плате, посадочных мест.

Bill of Materials (перечень элементов или спецификация) — вызывается из главного меню "Инструменты / Перечень элемен- тов / ASCII Output" — документ не должен содержать сообщений об ошибках.

Electrical Rules Check Report — вызывается из главного меню "Инструменты / Проверка электрических правил". В идеале в конце должны быть строки:

Netlist generated OK. (Список цепей сгенерирован.) No ERC errors found. (Ошибки в схеме не найдены.)

Однако не все выявленные ошибки действительно являются таковыми, в частности, к ним система относит незадействован- ные входные контакты компонентов.

Для передачи информации в программу проектирования пе- чатных плат выбирается пункт меню "Инструменты / Список связей в ARES" или кнопка на панели инструментов  .

### Проектирование печатной платы

##### Загрузка исходных данных

Проектирование осуществляется в программе ARES, кото- рую рекомендуется запускать из ISIS (в последних версиях среды программы работают в общем окне). Обе программы поддержи- вают так называемых "живой" список соединений, когда измене- ния в компонентах, соединениях, корпусах и т.д., произведенные в одной программе, немедленно передаются в другую.

При первом запуске выводится запрос о выборе шаблона для печатной платы, по умолчанию предлагается "DEFAULT".

Активизация значка  на левой инструментальной панели выводит в окно селектора объектов список всех компонентов, пока еще не размещенных на плате. Выделяя поочередно компоненты списка, убедиться, что для всех существуют посадочные места, изображения которых выводятся здесь же.

В процессе проектирования рекомендуется периодически сохранять результаты (файл платы имеет расширение .LYT).

##### Конфигурация платы

Первым шагом является создание контура печатной платы. Активизировать значок  на левой вертикальной панели. В селекторе слоев выбрать "Board Edge":  . Установить курсор мыши в точку предполагаемого левого верхнего угла платы, кликнуть левой кнопкой, нарисовать контур и еще раз нажать левую кнопку. В принципе можно создать контур любой формы, используя дуги и полилинии.

##### Компоновка

На следующем этапе необходимо разместить элементы на плате. Хотя в программе и существует автоматический режим (главное меню: "Инструменты / Авторазмещение"), пользоваться им рекомендуется только после ручного размещения тех компо- нентов, которые должны быть привязаны к определенным пози- циям. В настоящем проекте в первую очередь необходимо рас- ставить:

* монтажные отверстия (можно использовать встроенную заго- товку: металлизированные отверстия диаметром 3 мм, для удаления металлизации выбрать из контекстного меню пункт "Изменить слой / Drill Hole");
* соединители (обычно по краям платы и с определенной ориен- тацией);
* переменные резисторы (исходя из эргономики);
* мощные выходные транзисторы (для возможности установки на радиаторы охлаждения).

Для ручного размещения выбирается строка из списка ком- понентов, в нужном месте на плате делается сначала одинарный щелчок, компонент приобретает зеленый цвет, в это время его можно двигать, поворачивать (клавишами "+", "-" ), еще один щелчок фиксирует компонент, при этом он исчезает из списка. В зависимости от состояния переключателя слоев элементы можно размещать как на верхней стороне платы ("Component Side"), так и на нижней ("Solder Side"). Позицию компонента можно зафик- сировать так, чтобы случайно не сдвинуть его при последующих манипуляциях, для этого необходимо щелкнуть по объекту пра- вой кнопкой мыши, из выпадающего меню выбрать "Правка свойств", отметить опцию "Блокировать позицию".

Электрические связи между компонентами обозначены тон- кими зелеными линиями, а желтые линии со стрелками указыва- ют на более оптимальное с точки зрения системы направление размещения компонента. Посадочное место (а также независимо и составляющие его части — контактные площадки, обозначе- ния) на плате можно перемещать, вращать, редактировать. Уда- ленный с платы элемент снова появится в списке.

Расстановку компонентов удобно контролировать с помо- щью 3D-изображения узла (меню "Вывод / 3D визуализация").

##### Трассировка

В данной работе предлагается воспользоваться режимом ав- томатической трассировки, выбрав пункт меню "Инструменты / Автотрассировка", согласиться с параметрами по умолчанию. Убедиться, что в результате разводки питающие шины практиче- ски не отличаются от сигнальных.

Отменить произведенную разводку можно стандартным от- катом (Ctrl+Z), а также выделив и удалив дорожки. Для послед- ней операции используется

фильтр выбора, расположенный внизу окна. Выделив все элемен-

ты на печатной плате, оставить в фильтре активными две кнопки с подсказками "Выбрать дорожки", "Выбрать переходные отвер- стия", нажать клавишу Delete.

Изменить стратегию, задав увеличенную ширину проводни- ков для земли и шин питания. Вызвать диалог "Технология / Пра- вила проекта". На вкладке "Классы связи" для класса POWER задать иной стиль разводки — трассу шириной не менее 2.5 мм и увеличенный диаметр переходных отверстий. В Proteus ширина обозначается именем Txx, где xx — размер в сотых долях дюйма: например, T10 соответствует 0.254 мм.

Также предлагается освоить такой прием, как задание обла- сти, в которой запрещена прокладка проводников (например, это могут быть зоны установки металлизированных объектов — ре- зонаторов, радиаторов и т.п.). Для формирования области на ле- вой панели инструментов выделить значок "2D графика…" в виде прямоугольника, круга или полилинии, в селекторе слоев вы- брать  , разместить зону на плате.

Осуществить повторную трассировку. Проверить проект на отсутствие ошибок, которые индицируются в нижней строке ок- на, например,  . Щелчок по данному сообщению открывает протокол с более подробным описанием. Принять ме- ры к устранению ошибок.

Проверить функцию визуализации отдельных слоев с помо- щью диалога, вызываемого значком на верхней панели  или пунктом меню "Вид / Слои".

Получить 3D-изображение печатного узла ("Вывод / 3D ви- зуализация").

Продемонстрировать результат преподавателю. При отсут- ствии замечаний **привести в отчете**:

* полное изображение печатной платы;
* рисунки отдельно верхнего и нижнего слоев;
* 3-4 трехмерных изображения, включая плату без компонентов;
* основные нормы проекта: ширины дорожек, минимальные за- зоры между контактными площадками и трассами.

## Содержание отчета

Схемы устройств демонстрационных примеров (иметь в от- чете или отдельно в отпечатанных методических указаниях), схема исследуемого устройства.

Диаграммы, таблицы, скриншоты (перечислены в каждом пункте выполнения работы). Иллюстрации должны иметь заго- ловки, диаграммы — обозначения осей и всех кривых.

## Библиографический список

1. Proteus по-русски [Электронный ресурс] // Радиоежегод- ник, 2013, вып. 24. — Режим доступа: [http://www.rlocman.ru/radioyearbook/.](http://www.rlocman.ru/radioyearbook/)
2. Гололобов В.Н. ISIS. Руководство пользователя [Элек-

тронный ресурс]. — Режим доступа: [http://vgololobov.narod.ru/.](http://vgololobov.narod.ru/)

1. Файлы справок среды Proteus 8: ISIS.chm, ISISTUT.chm, LISA.chm, ARES.chm, ARESTUT.chm (русскоязычный перевод В.Данилова, 2013) [Электронный ресурс]. — Режим доступа: [http://kazus.ru/forums/attachment.php?attachmentid=51457&d=1378051358,](http://kazus.ru/forums/attachment.php?attachmentid=51457&d=1378051358) [http://kazus.ru/forums/attachment.php?attachmentid=69139&d=1409503333,](http://kazus.ru/forums/attachment.php?attachmentid=69139&d=1409503333) [http://kazus.ru/forums/attachment.php?attachmentid=53929&d=1383050589.](http://kazus.ru/forums/attachment.php?attachmentid=53929&d=1383050589)

## Контрольные вопросы

###### Общие вопросы

* Охарактеризуйте назначение и возможности программы.
* Дайте сравнительную оценку систем проектирования Multisim, OrCAD, Proteus.
* Какие виды проектов существуют в среде, каковы их основ- ные настройки? Какие файлы входят в состав различных про- ектов?
* Опишите типы библиотек компонентов (раскройте деление на категории, подкатегории, производителей).
* Опишите типы библиотек корпусов (раскройте деление на ка- тегории, типы, подкатегории).
* Охарактеризуйте виртуальную измерительную аппаратуру.
* Расскажите об элементах интерактивной отладки (органы управления, отображения, средства анимации).

###### Подготовка схем

* Опишите последовательность выбора компонента из библио- теки и размещения его на поле схемы.
* Сравните способы соединения компонентов: проводниками, терминалами, линиями групповой связи.
* Расскажите об особенностях подключения цепей питания.
* Какие позиционные обозначения компонентов используются в программе, а какие приняты в ЕСКД? Каковы способы расста- новки обозначений.
* Как редактировать свойства компонентов?
* Как создать новый компонент?
* Как организовать многостраничный (иерархический) проект?
* Опишите способы верификации проекта.
* Какие схемотехнические ошибки могут распознаваться систе- мой проектирования?

###### Учебные проекты

* Как узнать текущие напряжения и токи в цепях без подключе- ния измерительных приборов?
* Нарисуйте диаграммы напряжения источника, напряжения на базе транзистора, тока через транзистор в проекте Demo-AC.
* Приведите количественные параметры генераторов сигналов: диапазон частот, диапазон напряжений, типы сигналов.
* Каковы правила работы с логическим анализатором?
* Назовите функции всех компонентов в проекте Demo-Clock.
* Раскройте принцип управления скоростью и направлением вращения в проекте Demo-Motor. Нарисуйте диаграммы напряжения на выводах мотора.
* Поясните принцип регулирования уровней низких и высоких частот в пассивном регуляторе тембра. Как получить линей- ную АЧХ при среднем угле поворота ротора переменного ре- зистора?
* Какими элементами задается коэффициент передачи и рабочая точка усилителя в проекте SoundAmplifier?
* Можно ли источник многочастотного сигнала реализовать на основе источников напряжения (проект SoundAmplifier)?

###### Методы моделирования

* Каковы особенности проведения анализа статического режи- ма?
* Что представляют собой результаты анализа переходных про- цессов? Какие исходные данные необходимы для его проведе- ния?
* В чем различие между такими видами анализа, как аналого- вый, цифровой, смешанный?
* Как сдвинуть, смасштабировать и произвести дополнительную обработку отдельных кривых на диаграмме?
* Каков алгоритм проведения частотного анализа? Какой вход- ной сигнал используется в каждый момент времени?
* Как получить частотную зависимость таких параметров, как входной и выходной импедансы?
* Какие элементы схемы участвуют при проведении анализа шумов? Что такое выходной шум и шум, приведенный ко вхо- ду? Какие единицы измерения используются при построении графиков шумов?
* Объясните причины появления искажений сигнала, какими элементами схемы они обусловлены? Что такое интермодуля- ционные искажения?
* В чем смысл преобразования Фурье? Каковы исходные дан- ные для проведения данного анализа? Объясните изменение спектральной характеристики при увеличении уровня входно- го сигнала.
* Какие параметры можно изменять при проведении вариацион- ных анализов на постоянном и переменном токе? Как произ- водится выбор переменной и задание количественных дан- ных?
* Что может являться результатом вариационного анализа на постоянном и переменном токе?

###### Проектирование печатной платы

* Укажите необходимые исходные данные для начала проекти- рования платы. Как исключить компоненты, предназначенные только для моделирования, из проекта печатной платы?
* Что такое шаблон печатной платы, какие шаблоны имеются в системе?
* Какие параметры входят в состав проектных норм, как они задаются?
* Как сформировать контур печатной платы сложной конфигу- рации?
* Какие правила существуют при размещении компонентов на печатной плате? В каких случаях используется автоматическая расстановка? Как заблокировать изменение позиции элемента?
* Как настроить правила для различных классов цепей (ширина,

сужение, переходные отверстия)?

* Как сформировать область запрета для размещения проводни- ков?
* Как запустить трассировку, как ее отменить, как осуществить ручную коррекцию топологии проводников?
* Как создать шину земли и/или питания с максимальной залив- кой площади платы?
* Как визуализировать отдельные слои?

**Иллюстрации к лабораторной работе**

Проект **Demo-AC**

|  |
| --- |
| ***Анимация цепей переменного тока*** |
|  |  |  |  |  |  | **HL2** |  |  |
|  |  |  |  |  |  | 12V |  |  |
|  |  | **HL1** | **T1** |  |  |  |  | **G2**12V |
|  |  | 220V |  |  | **R1** |  | **VT1** |  |
| **G1** | **PV1** |  |  | **PV2** | 500R |  |  |  |
|  | +88.8*Volts* |  |  | +88.8*Volts* |  |  |  |  |
| Напряжение источника переменного тока G1 с напряжением 220 В понижается трансформатором с коэффициентом трансформации 100:1 и используется для управления транзисторным ключом. |

Проект **Demo-FigLissajous**



|  |
| --- |
| ***Фигуры Лиссажу*** |
| **+** AB**-** C**AM FM**D**+****-****AM FM** |
| Проект демонстрирует наблюдение фигур Лиссажу на экране осциллографа в режиме X-Y |

Проект **Demo-Clock**

10Hz

**U10:A**

+5V

7420

Note how it is possible to create realistic looking display peripherals using our Active Circuit Technology.

**U8**

D0 D1 D2 D3

Q0 Q1 Q2 Q3 RCO

**U9**

D0 Q0

D1 Q1

D2 Q2

D3 Q3

RCO

ENP ENT CLK LOAD MR

74160

ENP ENT CLK LOAD MR

74160

**U2**

A B C D

BI/RBO RBI

LT

7447

QA QB QC QD QE QF QG

**R1**

4k7

**U10:B**

7420

**SW2**

**U7:A**

**U1**

A B C D

BI/RBO RBI

LT

7447

QA QB QC QD QE QF QG

**SET HOURS**

7410

**U5**

D0 D1 D2 D3

Q0 Q1 Q2 Q3 RCO

**U6**

D0 Q0

D1 Q1

D2 Q2

D3 Q3

RCO

ENP ENT CLK LOAD MR

74160

ENP ENT CLK LOAD MR

74160

**U4**

A B C D

BI/RBO RBI

LT

7447

QA QB QC QD QE QF QG

**U7:C**

7410

**R2**

4k7

**U3**

A B C

 D BI/RBO RBI

LT

QA QB QC QD QE QF QG

**SW1**

**U7:B**

7447

**SET MINS**

7410

**U13**

D0 D1 D2 D3

Q0 Q1 Q2 Q3 RCO

**U11**

D0 D1 D2 D3

Q0 Q1 Q2 Q3 RCO

**U12**

D0 D1 D2 D3

Q0 Q1 Q2 Q3 RCO

***ЦИФРОВЫЕ ЧАСЫ НА TTL-ЛОГИКЕ***

Проект цифровых часов с кнопками предустановки значений часов-минут. Симуляция в реальном времени !

ENP ENT CLK LOAD MR

74160

ENP ENT CLK LOAD MR

74160

ENP ENT CLK LOAD MR

74160

Используется моделирование в смешанном режиме. Цифровые устройства: декадные счетчики, дешифраторы семисегментного кода, логические элементы; аналоговые: 7-сегментные индикаторы.

**?**

Проект **Demo-Motor**

C1

10pF

R1

1M

X1

1MHz

C2

30

29

27

U1

XTAL EXTAL E

PB[0..7]

PB[0..7]

PC[0..7]

10pF

PC[0..7]

25

24 MODA/LIR

MODB

STRA 26

STRB 28

41

40

39

+12V

+5V

INPUT

IRQ XIRQ RESET

 SW1

RV1

SW-SPDT

17

18

19

20

PE0/AN0 PE1/AN1 PE2/AN2 PE3/AN3

PA0/IC3 PA1/IC2 PA2/IC1 PA3/OC5/OC1 PA4/OC4/OC1 PA5/OC3/OC1 PA6/OC2/OC1 PA7/PAI/OC1

8

7

6

5

4

3

2

1

R2

1k

R5

1k

OC2 OC1

Q8

Q7

BC184

22

21

1k

PD0/RxD PD1/TxD PD2/MISO PD3/MOSI PD4/SCK PD5/SS

42

43

44

45

46

47

Q4

Q1

TIP31

BC184

VRH VRL

MC68HC11A8

PROGRAM=PWMMOTOR.S19

R4

1k

TIP31

R7

1k

Q5

Q6

Q3

BC184

Q2

TIP32

***Контроллер привода с ШИМ***

В проекте демонстрируется контроллер с широтно-импульсной модуляцией на базе микропроцессора семейства HC11, работающего по программе, написанной на языке ассемблера.

TIP32

BC184

Используется программный ввод уровня сигнала через канал АЦП микропроцессора и генерация ШИМ сигнала с помощью встроенного таймера.

На выходной усилитель мощности (мост Дарлингтона)

с процессора поступают: ШИМ сигнал для задания скорости вращения мотора и сигнал выбора направления вращения. При смене направления изменяется полярность напряжения импульсов, подаваемых на мотор.

**?**

**?**

+88.8

PB7 PB6 PB5

PB4

PB3 PB2 PB1

PB0

PC7 PC6 PC5

PC4

PC3 PC2 PC1

PC0

Проект **SoundAmplifier**

+12V +12V

J1

1

2

Вход

R1

4.7K

R10

680

C1

R5

C3

0.015u

1

Q1

TIP31

**Источник многочастотного сигнала**

0.022u

4.7K

D1

1N4448HWT

R12

0.22

F1

R2

3

47k

R4

R7

3

47k

-12V

J2

3.3K

C6

C2

R6

1

2

F2

0.22u

0.22u

470

D2

1N4448HWT

R13

0.22

Выход

F3

R3

470

C4

0.15u

U1

 3

6

1

 2

Q2

TIP32

AD711KS

RS

1

C5

R11

22u

C7

20K

R8

470K

R9

2.0K

0.22u

A

-12V

B

C D

J3

+12 V

2

J4

GND

J5

-12 V

10%

1

2

9%

1

2

 4

 7

2

3

3

**Лабораторная работа № 4** (редакция 09.2015, составитель А.Н. Богаченков, ©МИРЭА, каф. РСПИ) **Основы компьютерного проектирования РЭС**

**ПРОГРАММИРУЕМЫЕ УСТРОЙСТВА В СРЕДЕ PROTEUS**

Работа посвящена знакомству с такими классами устройств, как управляемые и программируемые источники сигналов, ин- терфейсные компоненты, микропроцессоры.

## Источники сигналов

Источник сигнала может быть выбран из списка, активируе- мого значком  на левой вертикальной панели. Такие источники на схеме отображаются в виде символа  , указывающего на заданную цепь, второй вывод всегда связан с общим проводом GND. Источник может быть сконфигурирован как аналоговый (с постоянным уровнем, синусоидальный, импульсный, экспонен- циальный, частотно-модулированный и др.) и цифровой. Послед- ний обычно работает с двумя выходными уровнями ("0", "1"), но некоторые типы имитируют также 3-е состояние, "сильные" и "слабые" логические уровни. Цифровые источники могут генери- ровать тактовую частоту, одиночные фронты и импульсы, а так- же сложную программируемую последовательность импульсов.

Аналогичные источники, но уже в виде двухполюсников располагаются в библиотеке компонентов: Simulator Primitives / Sources.

В составе среды также имеются управляемые источники то- ка/напряжения, являющиеся четырехполюсниками — размеща- ются в библиотеке Modelling Primitives / Analog (SPICE):



AVCVS (Arbitrary Voltage Controlled Voltage Source) –

нелинейный источник напряжения, управляемый напряжением; AVCCS (Arbitrary Voltage Controlled Current Source) –

нелинейный источник тока, управляемый напряжением; ACCVS (Arbitrary Current Controlled Voltage Source) –

нелинейный источник напряжения, управляемый током; ACCCS (Arbitrary Current Controlled Current Source) –

нелинейный источник тока, управляемый током.

Передаточная функция (Transfer Function) может использо- вать любые математические операции и функции: SIN, SINH, ASIN, ASINH (синусы); COS, COSH, ACOS, ACOSH (косинусы);

TAN, TANH, ATAN, ATANH (тангенсы); ABS (абсолютное зна- чение); EXP (экспонента); LN, LOG (натуральный и десятичный логарифмы); SQRT (квадратный корень); LIMIT (предельные значения); PWR, PWRS (степень); U, STP, URAMP, SGN. Вход- ными аргументами являются переменные A, B (управляющие входные уровни).

Например, если к каждому входу A и B компонента AVCVS подключить источники напряжения с единичными амплитудами и частотами fA > fB, то функция V(A)\*(0.3\*V(B)+1) сформи- рует на выходе амплитудно- модулированный сигнал с 30%-ным коэффициентом модуляции.

В той же библиотеке Modelling Primitives / Analog (SPICE) имеются и простые линейные источники VCVS, VCCS, CCVS, CCCS, в которых задается фактически один параметр — коэффи- циент передачи. Такие источники, в частности, можно использо- вать как макромодель усилителя.

## Микропроцессоры

Система проектирования Proteus Design Suite поддерживает симуляцию и отладку проектов на основе микропроцессоров и программируемых логических матриц. Среда содержит модели следующих семейств микроконтроллеров: PIC, AVR, 8051, HC11, ARM7, ARM Cortex-M3, MSP430, F2802x. Их список постоянно

расширяется с выходом новых версий программы. Поддержива- ется интерактивная отладка всей микропроцессорной системы, в ряде случаев — в режиме реального времени. Реализованы стан- дартные отладочные приемы: точки останова (включая различ- ные условия), пошаговое выполнение, просмотр внутренних ре- сурсов — переменных, регистров, памяти. Непосредственно в состав пакета входит ограниченный набор компиляторов с языка ассемблера (PIC, AVR, 8051, HC11), однако среда может взаимо- действовать с большинством профессиональных инструментов разработки программного кода на языках ассемблера и Си (Keil uVision, IAR Embedded Workbench, Microchip MPLAB, Atmel Studio и др.). В Proteus также встроены средства отладки комму- никационных интерфейсов RS-232, SPI, I2C, USB.

При симуляции микропроцессор рассматривается как чисто цифровое устройство, например, для выходных сигналов не ана- лизируется нагрузочная способность, не моделируются внешние цепи синхронизации (тактовая частота задается в свойствах ком- понента). Загружаемая программа представляется в виде двоич- ного исполняемого кода (обычно используется формат Intel Hex с файловым расширением .hex). Для отладки на уровне исходного текста нужен сам текст (размещается в каталоге с проектом), а также отладочная информация об адресах, номерах строк, пере- менных и др. Отладочный формат (ELF, DWARF, UBROF, OMF и др.) создается компилятором при определенной его настройке. При наличии внешнего компилятора рекомендуется воспользо- ваться еще одним дополнением к системе Proteus: интегрирован- ной средой разработки VSM Studio, которая облегчает взаимо- действие ISIS со многими существующими компиляторами.

## Порядок выполнения

Перед началом работы выполнить процедуру «**Инициализа- ция Proteus**», используя соответствующий значок на рабочем столе. Запустить программу ISIS.

### Моделирование генераторов сигналов

##### Моделирование последовательного интерфейса

Создать и проверить работу генератора тестовой цифровой последовательности импульсов в соответствии со стандартным протоколом RS-232. При передаче по данному интерфейсу ис- пользуется одна сигнальная линия с исходным уровнем лог. "1". Байт состоит из 10 посылок: стартовый бит "0"; 8 информацион- ных бит данных, начиная с младшего; стоповый бит "1". Пример временной диаграммы приведен ниже:



Разместить на схеме генератор типа DPATTERN (или любой из списка, задав в его свойствах цифровой тип "Шаблон"). Согла- совать с преподавателем генерируемую информацию (например, строку вида "ab\_"). Перевести ASCII коды символов заданной строки в битовую последовательность (например, для 3-х симво- лов длина составит 30 бит). Фрагмент ASCII таблицы приведен ниже — код символа равен сумме кодов соответствующих строки и столбца, значения даны в16-ричной системе (H – суффикс).



Дополнить последовательность еще десятком или более еди- ниц с целью визуального разделения пачек импульсов на экране осциллографа.

Пример. Строка "A+B" состоит из символов с кодами 4116, 2B16, 4216. В следующих строках приведены а) двоичные коды символов; б) перевернутые коды символов (младший бит слева); в) коды со стартовыми, стоповыми и разделительными битами:

а) 0100 0001 0010 1011 0100 0010

б) 1000 0010 1101 0100 0100 0010

в) **0**1000 0010**1 0**1101 0100**1 0**0100 0010**1** 1111111111

Битовую строку (без пробелов) ввести в поле "Определенная последовательность импульсов" свойств генератора или, после нажатия кнопки "Правка", сформировать непосредственно в виде диаграммы.

Для корректности проведения моделирования рекомендуется генератор нагрузить на активное сопротивление.

Для регистрации процесса подключить к генератору прибо- ры: осциллограф и терминал (Virtual Terminal, вход RXD). Задать длительность импульса генератора, соответствующую скорости регистрации терминала (параметр Baud Rate с размерностью бит/с). Проверить работу созданной модели.

Оценить, какое максимальное изменение скорости (или дли- тельности передачи бита) не нарушает корректное распознавание принимаемой информации. **Привести в отчете**: схему моделиро- вания, текст заданной строки, битовую строку, диаграмму сигна- ла, исходные значения длительности бита генератора и скорости приема, предельно допустимую длительность бита или скорость (в т.ч., отклонение в процентном отношении).

##### Генерация модулированного сигнала

Дополнить предыдущий проект двумя генераторами синусо- идального сигнала и двумя нелинейными источниками тока, управляемыми напряжением. Выставить частоты генераторов так, чтобы за время передачи бита один из них формировал 2. 5

периодов, а второй, например, в 2 раза больше. Подключить каж- дый из генераторов синуса к управляющему входу своего нели- нейного источника, а на вторые управляющие входы обоих ис- точников подать битовую последовательность. Выходы нелиней- ных источников тока запараллелить на общий резистор и подве- сти ко второму входу осциллографа. Задать передаточные функ- ции источников с целью осуществления частотной манипуляции (по умолчанию лог. "1" соответствует уровню напряжения 5 В). Проверить работу устройства, продемонстрировать преподавате- лю, **привести в отчете** схему устройства, диаграмму полученно- го ЧМ-сигнала.

##### Генератор с кусочно-линейной формой сигнала

Начать новый проект. Разместить в схеме генератор типа PWLIN (или любой из списка, выбрав в его свойствах аналоговый тип "Настраиваемый"). Задать максимальное время (XMax) не более 1 c (лучше несколько миллисекунд), на графическом поле создать произвольную форму кривой (не менее, чем по 5 точкам). Подключить к выходу генератора осциллограф. Запустить моде- лирование.

Так как формирование сигнала осуществляется однократно, при настройках осциллографа по умолчанию (циклическая раз- вертка) сигнал возможно не будет визуализирован. Остановить моделирование, снова запустить кнопкой "Шаг"  , в от- крывшемся окне осциллографа переключить режим запуска из "Auto" в "One-Shot" (ждущий однократный), продолжить модели- рование в непрерывном режиме, при необходимости отрегулиро- вать развертку и усиление осциллографа.

Продемонстрировать преподавателю и **привести в отчете**

полученную осциллограмму и заданный шаблон сигнала.

##### Скриптовый генератор

Генератор позволяет формировать аналоговые или цифровые сигналы с помощью простого языка программирования, называе- мого EasyHDL.

Загрузить демонстрационный проект

##### D:\Proteus\Lab2\Demo-QPSK Modulation.DSN

Проверить работу. Вызвав свойства генератора, ознакомить- ся с текстом скрипта. Скопировать текст в буфер.

Создать новый проект с генератором и осциллографом. Для генератора задать аналоговый тип "EasyHDL", дать имя генера- тору, вставить текст скрипта. Убедиться в работоспособности. Упростить скрипт для получения только одного гармонического сигнала. Выполнить иную модификацию по заданию преподава- теля. **Привести в отчете** схему, текст скрипта, осциллограммы сигнала.

### Моделирование микропроцессорной системы

Для лабораторной работы взята схема реальной отладочной платы на микроконтроллере семейства MCS-51 (8051/52). Пери- ферийными устройствами являются 4 датчика-переключателя, 4 кнопки, 8 светодиодных индикаторов, жидкокристаллический символьный дисплей (LCD) на 16 x 2 символов, имеется также выход на звуковой излучатель. Для подготовки и компиляции программы на языке Си используется среда разработки Keil uVision с пакетом C51.

##### Взаимодействие с периферийными устройствами

Открыть проект

##### D:\Proteus\Lab2\HW-SPAR3A-8051.DSN

Сопоставить все периферийные устройства, показанные на схеме, с реальным макетом.

Из меню "Исходник" вызвать текст рабочей программы Prog1.c. Если комментарии на кириллице отображаются некор- ректно, выполнить шаг моделирования (  ), текст откроется в окне "8051 CPU Source Code". По комментариям понять произ- водимые действия. Запустить/продолжить моделирование. Про- верить работу всех органов управления и отображения.

Поставить процесс на паузу (  ), из меню "Отладка" от- крыть все еще неоткрытые окна "8051 CPU …". Используя два режима пошаговой работы — клавишей F10 и кнопкой  , отметить, какого рода шаги реализуются в этих режимах. **Приве- сти в отчете** краткое описание содержимого отладочных окон (хотя бы перевод на русский язык заголовков окон).

С помощью виртуальных осциллографа и/или таймера- счетчика измерить и **привести в отчете** минимальную и макси- мальную частоты генерируемого сигнала. Удалить приборы из проекта (с целью сохранения высокой скорости моделирования).

Проверить те же действия на реальной плате: включить ла- бораторный макет, запустить программу **Lab51** соответствую- щим значком на рабочем столе. Программный переключатель  поставить в положение ICE. Убедиться, что на справоч- ной схеме макета присутствуют все моделируемые периферий- ные устройства. С помощью меню "Файл / Открыть машинный код" загрузить исполняемый код:

##### D:\Proteus\Lab2\Prog1\Prog1.hex.

Запустить на выполнение значком  .

Одновременно наблюдая за работой модели и живого устройства, качественно оценить различие в скорости работы.

При наличии осциллографа измерить те же минимальную и максимальную частоты сигнала, генерируемого макетом. **Приве- сти в отчете** реальные осциллограммы на предельных частотах. Остановить работу устройства кнопкой  .

Загрузить среду разработки **Keil uVision** соответствующим значком на рабочем столе. Из меню "Project / Open project" от- крыть проект: **D:\Proteus\Lab2\Prog1\Prog1.uvproj**.

Внести некоторые коррекции в рабочую программу, напри- мер: изменить характер взаимодействия кнопок и индикаторов; изменить частоты генерируемых тонов; сформировать музыкаль- ную фразу и т.п. (согласовать с преподавателем). Откомпилиро- вать программу через меню "Project / Build…" или клавишей F7. Убедиться в отсутствии ошибок. Для проверки работы в про- грамме Proteus необходимо перезапустить моделирование, в Lab51 — повторно загрузить исполняемый код. Продемонстри- ровать работу преподавателю.

Используя таблицу полупериодов в тексте программы, рас- считать максимальную частоту генерируемого сигнала, сравнить ее с измеренной величиной.

**Привести в отсчете** исходный текст программы (дан в при- ложении), отметить в нем произведенные изменения, привести расчет максимальной частоты. Рекомендация: для сохранения исходного форматирования текста программы его следует распе- чатать непосредственно из файла (.pdf), внеся затем рукописные коррекции.

##### Работа с символьным дисплеем

Загрузить в микропроцессор другую программу — в среде Proteus вызвать свойства процессора и в поле "Program File" ука- зать "Prog2\Debug.OMF". Проверить работу модели и реального устройства по этой программе, еще раз сравнить скорости.

Разместить на схеме модели логи- ческий анализатор для снятия диа- граммы обмена с дисплеем. Так как данные являются 4-разрядными, а ши- на связи с дисплеем 7-разрядная, то рекомендуется показанный на рисунке вариант сопряжения шин устройства и анализатора. К анализатору необходи- мо подвести и стробирующий сигнал

LCD–E (можно также добавить сигналы RS, RW). Принципы работы с анализатором изложены в предыдущей лабораторной работе. После старта моделирования нажать на паузу, задать пе- риод регистрации "Capture Resolution" порядка 1 мкс (это соот- ветствует длительности машинного цикла моделируемого про- цессора), задать запуск от сигнала LCD–E по фронту, регулятор позиции поставить в положение примерно 50%, возобновить ра- боту, нажать "Capture" на анализаторе, поставить процесс моде- лирования на паузу, масштабировать диаграмму для наблюдения двойного стробирующего импульса. **Привести в отчете** схему устройства с анализатором, временную диаграмму.

Скорректировать проект Keil uVision

##### D:\Proteus\Lab2\Prog2\Prog2.uvproj

с целью, например, вывода на дисплей заданного текстового со- общения. Продемонстрировать работу на любом устройстве. Оформление отчета по данному заданию не требуется.

В программе Keil uVision закрыть проект — "Project / Close project", закрыть Keil uVision, Lab51, выключить макет.

### Выбор и редактирование посадочных мест

Подготовить проект микропроцессорной системы для созда- ния печатной платы. В программе ISIS просмотреть список ком- понентов (меню "Проект / Проводник проекта"). В данном случае для анимированных и интерактивных компонентов отсутствуют посадочные места (помечены как "**missing**"). Кроме того, щелк- нув правой кнопкой на схеме по компоненту LCD и выбрав пункт "Корпус" можно увидеть, что посадочное место CONN-DIL14 (с двухрядным расположением выводов) не соответствует реально- му дисплею, установленному на лабораторном макете. Выбор и замена корпусов, имеющихся в библиотеках Proteus, могут быть проведены как в программе ISIS, так и ARES, а редактирование и создание новых — только в ARES.

Порядок назначения в ISIS: пункт "Корпус" контекстного меню компонента, далее "Добавить", выбор корпуса из библиоте- ки, назначение именам/номерам выводов схемного компонента (указаны в левом столбце "Пин") номеров/имен выводов корпуса (указаны в правом столбце "A"), кнопка "Назначить", сохранение в пользовательской библиотеке USERDVC (или иной), согласие на обновление устройства с созданным корпусом.

Выбрать корпус для светодиодов, например, Miscellaneous / Through Hole / LEDs / LED. Указанный корпус получают сразу все 8 имеющихся компонентов.

Выбрать корпус для переключателей: Miscellaneous / Through Hole / Switches / SW-SS-03. Задать номера: контакт COM (об- щий) – 2, NC (нормально замкнутый) – 1, NO (нормально разо- мкнутый) – 3. В графическом обозначении элемента на схеме появится новая нумерация контактов.

Выбрать корпус для кнопок: Miscellaneous / Through Hole / Switches / SW-FSM4. Этот корпус имеет 4 вывода, однако элек- трически вывод 1 соединен с 2, а вывод 3 — с 4, поэтому при назначении номеров для одного контакта схемного элемента ре- комендуется указывать оба корпусных (через запятую).

Однотипные компоненты, в которых изначально не совпада- ли обозначения контактов, не сразу получают новый корпус, хотя в свойствах он и указан (возможно это недоработка программы). Поэтому либо процедуру назначения необходимо повторить для всех экземпляров, либо (рекомендуется) — сохранить проект и снова его открыть.

Для редактирования посадочного места компонента LCD за- пустить программу ARES (можно как значком на рабочем столе, так и из ISIS). Активизировать список корпусов (кнопка  на левой панели), выбрать шаблон-заготовку. Если в списке ничего подходящего нет или он пуст, добавить из библиотеки (кнопка "P" или меню "Библиотека / Выбрать Корпус/Символ"). Для нашего дисплея заготовкой могут быть CONN-DIL14, CONN- DIL16, CONN-SIL18. Перенести корпус на рабочее поле. Выде- лив его, выбрать из контекстного меню или главного меню "Биб- лиотека" пункт "Разложить…". Компонент при этом распадается на отдельные графические примитивы и скриптовой текстовой блок. Стандартными графическими манипуляциями сформиро- вать требуемую конфигурацию из 16 контактных площадок. Ак- тивизировав слой "Top Silk", нарисовать внешние контуры дис- плейного модуля. Желательно также добавить монтажные отвер- стия.



Выделить общей рамкой созданный рисунок и скриптовой блок, выбрать пункт меню "Создать корпус". В следующем диа- логе ввести новое имя корпуса (например, LCD-SIL16), при же- лании изменить категорию, подкатегорию, скорректировать опи- сание, выбрать библиотеку (по умолчанию USERPKG). В этом же диалоге перейти на вкладку "3D визуализация", в окне "Парамет- ры модели" удалить все строки. По умолчанию для визуализации используется параллелепипед красного цвета с осевыми или пла- нарными выводами в зависимости от типа контактной площадки. Изучение формата описания 3D-модели выходит за рамки насто- ящей работы, можно попробовать только следующие строки:

**COLOUR=(0,100,0)** ; Цвет в формате (R,G,B)

**MINHEIGHT=2mm** ; Зазор плата–компонент

**MAXHEIGHT=8mm** ; Макс. высота над платой

Нажать OK для сохранения посадочного места в библиотеке.

Удалить с рабочего поля все части заготовки.

Изменение корпуса возможно в ARES на любой стадии ком- поновки и трассировки (достаточно указать в свойствах компо- нента новое название). Однако рекомендуется это сделать в ISIS: вызвать соответствующий диалог пунктом контекстного меню "Корпус"; удалить существующий корпус; добавить новый; **про- верить** и при необходимости скорректировать обозначения (ну- мерацию) выводов; нажать "Назначить"; подтвердить сохранение в библиотеке и обновление на схеме. Просмотреть список компо- нентов в проводнике проекта, **сохранить** проект.

### Проектирование печатной платы

Руководствуясь инструкциями из предыдущей работы осу- ществить ручную компоновку (по образцу реального макета или использовать собственный дизайн) и автотрассировку. Дополни- тельно разместить на плате несколько надписей (названия соеди- нителей, кнопок).

**Привести в отчете** 2D и 3D виды печатной платы.

## Содержание отчета

Указано в каждом из пунктов выполнения работы.

## Контрольные вопросы

1. Перечислите типы источников сигнала и все библиотеки, их содержащие.
2. Какие виды сигналов можно сформировать с помощью аналоговых источников? Для каждого из сигналов перечислите задаваемые исходные данные (настройки).
3. Изобразите фрагменты сигналов, которые можно сгенери- ровать с помощью цифровых источников.
4. Приведите компонентные уравнения для источников напряжения/тока, управляемых напряжением/током.
5. Изобразите схемы и приведите необходимые передаточ- ные функции для формирования, например, сигналов с фазовой модуляцией, сигналов с линейно-частотной модуляцией, Гауссов радиоимпульс, пачек сигналов с законом изменения амплитуды по экспоненте и др.
6. На основе управляемых источников разработайте макро- модели: усилителя с ограничением выходной амплитуды, усили- теля с автоматической регулировкой усиления, логарифмическо- го усилителя, детектора АМ сигналов и др.
7. Опишите стандарт передачи данных по интерфейсу RS-

232. Сформулируйте требования к стабильности скорости пере- дачи. Объясните причины искажения данных. Как идентифици- ровать ошибки при приеме?

1. Дайте подробные объяснения работы операторов языка

EasyHDL на примере одной из демонстрационных задач.

1. Перечислите известные семейства микроконтроллеров, ка- кие принципиальные отличия между ними существуют?
2. Какие форматы программы процессора используются в среде проектирования? Какими средствами можно осуществлять компиляцию программ?
3. Какие методы использует среда проектирования при мо- делировании схем на дискретных компонентах и работающих по программе? Объясните различие в скоростях работы модели и реального устройства.
4. Какие приемы предоставляет среда проектирования при отладке устройств на микропроцессорах?
5. Как настроить логический анализатор для регистрации одиночных импульсов, определенных кодов? Как осуществить регистрацию сигналов на заданном участке работы программы?
6. Как выбрать корпус/посадочное место для компонента? Каков порядок действий при отсутствии готового корпуса?

## Модель микропроцессорной системы

LCD1 LM016L

Vcc

Vcc

C2

C1

10uF

RP1

1

22pF

U1

XTAL1

C3

ZQ1

12MHz

19

R1

10k

18

XTAL2

22pF

P0.0/AD0 P0.1/AD1 P0.2/AD2 P0.3/AD3 P0.4/AD4 P0.5/AD5 P0.6/AD6 P0.7/AD7

39

38

37

36

35

34

33

32

P00 P01 P02 P03 P04 P05 P06

Vcc

P0[0..6]

P00 2

P01 3

P02 4

P03 5

P04 6

P05 7

P06 8

Vcc

9

RST

4.7k

SW3

SW2

SW1

SW0

Vcc

29

1. PSEN
2. ALE

EA

P2.0/A8 P2.1/A9 P2.2/A10 P2.3/A11 P2.4/A12 P2.5/A13 P2.6/A14 P2.7/A15

21

22

23

24

25

26

27

28

P20 P21 P22 P23 P24 P25 P26 P27

Vcc

R2 R3 R4 R5 R6 R7 R8 R9

330 330 330 330 330 330 330 330

Power

J1

1

2

3

4

5

6

7

8

P1.0/T2 P1.1/T2EX P1.2

P1.3 P1.4 P1.5 P1.6 P1.7

AT89C52 PROGRAM=Debug.OMF

P3.0/RXD P3.1/TXD P3.2/INT0 P3.3/INT1 P3.4/T0 P3.5/T1 P3.6/WR P3.7/RD

10

11

12

13

14

15

16

17

LD7 LD6 LD5 LD4 LD3 LD2 LD1 LD0

1

2

3

DS-261

BA1

SPEAKER

BA2

SPEAKER

J2

1

2

3

Phone

Jack 3.5

South

East

West

North

P27

1

2

3

VSS VDD VEE

P26

P04 P06 P05

4

5

6

RS RW E

P25

P24

7

8

9

10

11

12

13

14

D0 D1 D2 D3 D4 D5 D6 D7

P23

P00 P01 P02 P03

P22

P21

P20

# Файл Prog1.c

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

УЧЕБНЫЙ ПРОЕКТ для отладочной платы HW‐SPAR3A и симулятора Proteus

Устройство содержит микроконтроллер семейства MCS‐51 и следующие периферийные компоненты:

‐ 4 переключателя на разрядах порта P1.3...P1.0;

‐ 4 кнопки на разрядах порта P3.5...P3.2;

‐ 8 светодиодных индикаторов на разрядах порта P2.7...P2.0

‐ 2 звукоизлучателя на разрядах порта P3.7, P3.6

В исходном состоянии все органы управления формируют лог. "1", производится динамическое переключение светодиодных индикаторов с периодом, задаваемым пустым программным циклом. При воздействии на органы управления разрешается работа таймера, генерирующего запросы прерывания, периодичность запросов задается таблично.

В подпрограмме обслуживания прерывания осуществляется изменение уровня на звуковых выходах.

Некоторые пояснения по заданию частоты (периода) генерируемого сигнала. После сброса

таймер 2 микроконтроллера по умолчанию сконфигурован как автоперезагружаемый, с синхронизацией от внутренних тактовых импульсов (с частотой в 12 раз меньшей частоты кварцевого резонатора).

При частоте кварцевого резонатора 12 МГц временной интервал T в мкс получается при загрузке в регистровые пары (TH2,TL2) и (RCAP2H,RCAP2L) двухбайтового значения (65536 ‐ T).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <AT89X52.H> #define nt 65536

//Объявления регистров, адресов, констант

//Константа: макс. число тактов для таймера

//Таблица полупериодов тональных сигналов в мкс (для тактовой частоты 12 МГц)

**unsigned short** Tabl[] = { nt‐9091, nt‐2273, nt‐1136, nt‐568, nt‐284, nt‐239, nt‐159, nt‐71};

//ПОДПРОГРАММА ПЕРЕЗАГРУЗКИ ТАЙМЕРА (вход: индекс тонального сигнала)

**void** TimerReload(**int** t)

{

TR2 = 0;

TH2 = RCAP2H = Tabl[t] >> 8; TL2 = RCAP2L = Tabl[t];

TR2 = 1;

}

//Остановка таймера

//Загрузка нового значения полупериода из таблицы

// (по отдельности старшего и младшего байтов)

//Пуск таймера

//ГЛАВНАЯ ФУНКЦИЯ

**void** main(**void**)

{

//Объявление локальных переменных

**volatile int** i; //Переменная счетчика циклов **unsigned char** LEDState = 0x01; //Начальный код для индикаторов **unsigned char** KeyCode, KeyPrev = 0; //Код нажатия (текущий, предыдущий)

//Инициализационные действия

P2 = LEDState; ET2 = 1;

EA = 1;

//Бесконечный цикл основной программы

**while** (1)

{

//Начальное состояние светодиодных индикаторов

//Разрешение прерывания от таймера‐счетчика 2

//Общее разрешение прерываний

//Формирование кода нажатия от разрядов P1.3...P1.0, P3.5...P3.2

KeyCode = ~((P1 << 4)& 0xF0 | (P3 >> 2)& 0xF);

**if** (KeyCode) //ЕСЛИ КНОПКА/КЛЮЧ НАЖАТЫ

{

P2 = KeyCode;

**if** (KeyCode != KeyPrev)

{ KeyPrev = KeyCode;

**for** (i = 0;; i++)

{ **if** (KeyCode & 0x80) **break**; KeyCode <<= 1;

//Вывод состояния кнопок/ключей на индикаторы

//Если код нажатия не изменился, действий нет

//Здесь код нажатия новый, запоминание его

//Определение номера (0..7) нажатой кнопки

}

TimerReload(i); //Загрузка таймера новым кодом тона из таблицы

}

}

**else** //ЗДЕСЬ ВСЕ КНОПКИ/КЛЮЧИ ОТПУЩЕНЫ

{

KeyPrev = 0;

TR2 = 0;

**for** (i = 10000; i; i‐‐); LEDState = LEDState >> 1 |

//Запоминание текущего состояния кнопок/ключей

//Остановка таймера

//Некоторая программная временная задержка

//Изменение кода индикации

(LEDState & 1 ? 0x80 : 0);

P2 = LEDState; //Вывод нового кода на индикаторы

}

} //Конец тела оператора while (1)

} //Конец главной функции

//ОБРАБОТЧИК ПРЕРЫВАНИЯ ОТ ТАЙМЕРА‐СЧЕТЧИКА 2 (номер вектора прерываний = 5)

**void** Timer2Handler(**void**) interrupt 5

{

TF2 = 0;

P3\_6 = ~P3\_6; P3\_7 = ~P3\_7;

//Сброс флага запроса прерывания

//Инверсия разрядов звукового выхода

# Файл Prog2.c

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* УЧЕБНЫЙ ПРОЕКТ для отладочной платы HW‐SPAR3A и симулятора Proteus

\*

* Copyright (C) 2014 МГТУ МИРЭА

\*

* Устройство содержит микроконтроллер семейства MCS‐51,
* в проекте задействованы следующие периферийные компоненты:
* ‐ 8 светодиодных индикаторов на разрядах порта P2.7...P2.0;
* ‐ символьный дисплей, совместимый с контроллером HD44780:
* используется 4‐разрядная шина данных ‐ разряды порта P0.3...P0.0,
* сигналы RS ‐ P0.4, RW ‐ P0.6, E ‐ P0.5

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <AT89X52.H> //Объявления регистров, адресов, констант

#include <INTRINS.H> //Прототипы функций (\_nop\_ и др.)

#include "Prog2.h" //Определения внутренних функций

|  |  |  |  |
| --- | --- | --- | --- |
| #define | PortLCD | P0 | //Символическое имя для дисплейного порта |
| #define | Mask\_RS | 0x10 | //Байтовая маска на сигнал LCD RS |
| sbit | LCD\_E = | PortLCD^5; | //Символическое имя для сигнала строба LCD E |

**unsigned char** code LedTabl[][16]= //Массив кодов управления светодиодами

{ {0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01,0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01},

{0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80,0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80},

{0x18,0x24,0x42,0x81,0x81,0x42,0x24,0x18,0x18,0x24,0x42,0x81,0x81,0x42,0x24,0x18},

{0x3F,0x9F,0xCF,0xE7,0xF3,0xF9,0xFC,0x7E,0x3F,0x9F,0xCF,0xE7,0xF3,0xF9,0xFC,0x7E},

{0xFC,0xF9,0xF3,0xE7,0xCF,0x9F,0x3F,0x7E,0xFC,0xF9,0xF3,0xE7,0xCF,0x9F,0x3F,0x7E},

{0xAA,0x55,0xAA,0x55,0xAA,0x55,0xAA,0x55,0x0F,0xF0,0x0F,0xF0,0x0F,0xF0,0x0F,0xF0},

{0x80,0xC0,0xE0,0xF0,0xF8,0xFC,0xFE,0xFF,0x7F,0x3F,0x1F,0x0F,0x07,0x03,0x01,0x00},

{0x88,0x44,0x22,0x11,0x88,0x44,0x22,0x11,0x88,0x44,0x22,0x11,0x88,0x44,0x22,0x11}

};

//‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

// ГЛАВНАЯ ФУНКЦИЯ

**void** main(**void**)

{

//Объявление локальных переменных

**int** i, j; //Переменные счетчиков циклов

**unsigned char** sym = ' '; //Начальный выводимый символ (пробел)

//Инициализационные действия

P2 = LedTabl[0][0]; //Начальное состояние светодиодных индикаторов

LCD\_Reset(); //Инициализация дисплея

//Цикл основной программы

**while** (1)

{

LCD\_Cursor(0, 0); //Задание позиции курсора (строка, поз. в строке)

**for** (i = 0;i< 16; i++) //Цикл по позициям в строке дисплея

{ LCD\_Data(sym++); //Вывод символа с инкрементом его кода

**if** (sym < ' ') sym = ' '; //Пропуск пустых символов с кодами 0...0x1F

**for** (j = 5; j; j‐‐) //Цикл задержек между выводами символов

{ Delay60ms();

LedControl(); //Переключение светодиодных индикаторов

}

}

}

}

//‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

// УПРАВЛЕНИЕ СВЕТОДИОДАМИ

**void** LedControl(**void**)

{

**static char** m, n, p; //Индексы кода в строке, повторений строки, номера строки

**if** (!(m = ++m & 0x0F) && !(p= ++p& 0x03)) n= ++n& 0x07; P2 = LedTabl[n][m];

}

//‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

// ПОДПРОГРАММЫ ОБСЛУЖИВАНИЯ ДИСПЛЕЯ

// Установка позиции курсора:

// Row ‐ номер строки: 0 ‐ первая строка, 1 или иное ‐ вторая строка

// Column ‐ позиция в строке (0...15)

**void** LCD\_Cursor(**unsigned char** Row, **unsigned char** Column)

{

**if** (Row) LCD\_Write8(Column & 0x0F | 0xC0);

**else** LCD\_Write8(Column & 0x0F | 0x80);

// Вывод символа в текущую позицию

**void** LCD\_Data(**char** symbol)

{

PortLCD = symbol >> 4 & 0x0F | Mask\_RS; LCD\_E = 1;

\_nop\_(); \_nop\_(); //Здесь и далее пустые команды удлиняют импульс

LCD\_E = 0; // строба для удобства наблюдения

PortLCD = symbol & 0x0F | Mask\_RS; LCD\_E = 1;

\_nop\_(); \_nop\_();

LCD\_E = 0;

Delay40us();

}

// Инициализация

**void** LCD\_Reset(**void**)

{

PortLCD = 0; //Исходное состояние порта управления LCD

Delay20ms(); //Задержка перед началом инициализации

LCD\_Write4(0x03); // \*

Delay20ms(); // |

LCD\_Write4(0x03); // | Выдача определенной последовательности

Delay1ms(); // | кодов с определенными временными

LCD\_Write4(0x03); // | задержками через 4‐битный интерфейс

Delay40us(); // |

LCD\_Write4(0x02); // |

Delay40us(); // \*

LCD\_Write8(0x28); //Уст. 4‐битного интерфейса, формата символов

LCD\_Write8(0x06); //Настройка сдвига изображения и курсора

LCD\_Write8(0x0C); //Включение дисплея, гашение курсора

LCD\_Write8(0x01); //Очистка дисплея

Delay20ms();

}

// Передача в дисплей 4‐битного управляющего кода

**void** LCD\_Write4(**unsigned char** Code4)

{

PortLCD = Code4 & 0x0F; LCD\_E = 1;

\_nop\_(); \_nop\_();

LCD\_E = 0;

}

// Передача в дисплей 8‐битного управляющего кода

**void** LCD\_Write8(**unsigned char** Code8)

{

LCD\_Write4(Code8 >> 4 & 0x0F); LCD\_Write4(Code8 & 0x0F); Delay40us();

}

//‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

// ВРЕМЕННЫЕ ЗАДЕРЖКИ

**void** Delay40us(**void**) //40 мкс

{

**char** i= 17; **while** (‐‐i);

}

**void** Delay1ms(**void**) //1 мс

{

**char** i= 25; **do** Delay40us(); **while** (‐‐i);

}

**void** Delay20ms(**void**) //20 мс

{

**char** i= 20; **do** Delay1ms(); **while** (‐‐i);

}

**void** Delay60ms(**void**) //60 мс

{

**char** i= 60; **do** Delay1ms(); **while** (‐‐i);

}

**Лабораторная работа № 5** (редакция 11.2016, составитель А.Н. Богаченков, ©МИРЭА, каф. РСПИ) **Основы компьютерного проектирования РЭС**

**СИНТЕЗ ЦИФРОВЫХ УСТРОЙСТВ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА VHDL**

## Общие сведения

Лабораторная работа посвящена знакомству с методом со- здания цифровых устройств на основе программируемых логиче- ских интегральных схем (ПЛИС) с применением языка описания аппаратуры VHDL. Используется система автоматизированного проектирования (САПР) ISE Design Suite и отладочный модуль фирмы Xilinx.

Упрощенная функциональная схема аппаратного модуля приведена на рисунке 1. Все входные и выходные сигналы имеют логические уровни (0…3.3 В).

**BTN\_SOUTH**

push\_counter[2:0]

Рис. 2

**LED(0)**

**LED(1)**

**SW(0)**

**SW(1)**

24 Гц

**CLK**

3 Гц

**: 221**

**: 2**

**: 2**

**: 2**

50 МГц

Мульти- плексор clk24 freq

clk12 clk6

clk3

Демульти- плексор

Счетчик нажатий

##### .

**.**

##### .

**LED(7)**

**+**

Энкодер

**A B**

**FPGA**

 **BTN\_WEST**

**BTN\_SOUTH BTN\_EAST BTN\_NORTH**

**SW(0)**

**SW(1)**

**SW(2)**

**SW(3)**

**ROT\_A ROT\_B**

**LED(0)**

**LED(1)**

**LED(2)**

**LED(3)**

**LED(4)**

**LED(5)**

**LED(6)**

**LED(7)**

**AUD\_R AUD\_L**

правый

левый

Аудиовыход

**Generate Programming File** — создание файла для загрузки конфигурации в микросхему или запоминающее устройство.

Каждый этап в свою очередь состоит из нескольких проце- дур. Порядок выполнения процедур соответствует их расположе- нию в списке. Имеется возможность производить компиляцию по шагам, выбирая двойным щелчком требуемые процедуры. Акти- визация любой из них автоматически вызывает выполнение предыдущих.

Произвести компиляцию всего проекта выбором операции

##### Generate Programming File.

После успешного создания файла вызвать оболочку для про- граммирования, активизировав строку **Manage Configuration Project (iMPACT)**. В окне оболочки отобразится мнемосхема (рис. 3) с устройствами, которые потенциально можно програм- мировать: собственно ПЛИС (xc3a700a) и два вида ПЗУ:

**S ROT\_CENTER**

Генератор 50 МГц

**CLK**

## Порядок выполнения

Рис. 1

Рис. 3

Примечание. Если по каким-либо причинам мнемосхема не отобразилась, следует дважды щелкнуть на строке Boundary Scan, затем после щелчка правой кнопкой в основном окне выбрать пункт меню Initialize Chain, в последующих диалогах вы- брать для ПЛИС файл beg.bit, отказаться от конфигурирования других устройств.

При отладке проектов загрузка всегда производится непо-

###### Знакомство с программной средой. Реализация устрой- ства (Implementation)

Перед началом работы выполнить процедуру «**Инициализа- ция Multisim**», используя соответствующий значок на рабочем столе.

Запустить программную среду посредством значка на рабочем столе «**ISE Design Suite 14.7**». Открыть ознакомительный проект:

**D: \ PLIS \ CompDesign \ BegProject \ begproject.xise**

Левую часть рабочего поля занимает навигатор проекта с вкладками *Start*, *Design*, *Files*, *Libraries*. В лабораторной работе следует использовать только вкладку *Design*. Проверить, что пе- реключатель *View* в верхней строке установлен в положение *Im- plementation*, что соответствует реализации проекта в кристалле ПЛИС.

В состав проекта входит два файла:

**beg\_source.vhd** – описание синтезируемого устройства на языке VHDL: внешние сигналы, архитектура, функционирование; **s3ask.ucf** – файл проектных ограничений: соответствие сиг- налов выводам микросхемы, настройки входных и выходных

разрядов, временные константы и др.

Ознакомиться с описанием и функционированием устрой- ства по комментариям в файле и функциональной схеме рис. 2.

Компиляция проекта осуществляется в несколько этапов. При выделении в окне *Hierarchy* строки с файлом beg\_source.vhd перечень этапов появляется в окне *Processes*:

**Synthesize - XST** — синтез устройства в виде электрической схемы;

**Implement Design** — размещение логических элементов, трассировка соединений в кристалле;

средственно в ПЛИС, для чего правой кнопкой необходимо щелкнуть по значку микросхемы, выбрать из контекстного меню пункт **Program**. При успешной загрузке (сообщение "Program Succeeded") отладочный модуль сразу начинает работать в соот- ветствии с конфигурацией.

Проверить действие кнопки и переключателей. **Привести в отчете** краткое описание поведения индикаторов.

В тексте программы закомментировать операторы процесса подсчета нажатий кнопки без учета дребезга контактов, снять комментарии с операторов процесса с фильтрацией дребезга. Перекомпилировать проект (**Generate Programming File**), пере- ключиться на окно оболочки iMPACT, загрузить конфигурацию в отладочный модуль. Проверить действие кнопки, **отметить в отчете** различие в реакции индикаторов.

###### Симуляция поведения устройства (Simulation)

Освоить процесс симуляции функционального поведения синтезируемого устройства, используя проект предыдущего пункта.

ПЛИС являются быстродействующими устройствами, и си- муляция (моделирование) состояний обычно осуществляется с временным шагом порядка 1 пикосекунды. В рассматриваемом проекте для наблюдения выходных сигналов требуются времен- ные интервалы, исчисляемые секундами, моделирование при этом может занимать минуты работы компьютера. Поэтому в данном случае рекомендуется искусственное ускорение протека- ющих процессов.

Скорректировать оператор clk24 <= div\_clk(20), заменив его на clk24 <= div\_clk(10). Тем самым разрядность счетчика- делителя тактовой частоты уменьшается с 21 до 11, что соответ-

ствует увеличению внутренней тактовой частоты 24 Гц примерно в 1000 раз. В этом случае получаемые при симуляции времена в [мкс], [мс] для реального устройства следует интерпретировать как [мс], [с].

На вкладке *Design* в верхней строке изменить опцию *View* с *Implementation* на *Simulation*. Из проекта исчезает файл .ucf, но появляется файл **beg\_tb.vhd** . Данный файл является тестовым модулем, в котором имитируются все воздействия на моделируе- мое устройство, его также называют Test Bench ("испытательный стенд"). Головной файл описания устройства beg\_source.vhd ста- новится как бы внутренним модулем, он получает дополнитель- ное обозначение uut (Unit Under Test).

Шаблон тестового модуля генерируется системой проекти- рования, в нем необходимо только прописать последовательность формирования входных сигналов, в данном случае – уровней с органов управления и тактового сигнала (добавленные операции снабжены русскоязычными комментариями). Открыть и изучить тестовый файл beg\_tb.vhd.

Выделить строку **beg\_tb - behavior (beg\_tb.vhd)** в окне *Hier- archy*, раскрыть строку **ISim Simulator** в окне *Processes*, дважды щелкнуть по строке **Simulate Behavioral Model**. После некоторых проверок запустится программная утилита с названием iSim, в которой уже заготовлены диаграммы для всех входных и выход- ных сигналов. В левом столбце (*Instance and Process Name*) рас- крыть строку **beg\_tb**, выделить строку **uut**. В следующем столбце (*Simulation Objects for uut*) будет представлен список внешних и внутренних сигналов моделируемого устройства. Перетащить из этого списка на временную диаграмму внутренние сигналы **clk24**, **clk12**, **clk6**, **clk3**, **freq**, на временной диаграмме раскрыть код **led[7:0]**, в окошко с временем моделирования (по умолчанию выглядит как ) ввести значение **30ms**.

Для запуска моделирования нажать кнопку  , по оконча- нии которого кнопками  или колесиком мыши устано- вить удобный для наблюдения масштаб временной диаграммы.

**Привести в отчете** снимок (скриншот) диаграммы с полным циклом повторения входных сигналов.

Некоторые приемы работы в программе iSim.

Для повтора моделирования, например, после добавления новых сигналов на временную диаграмму, перезапустить процесс кнопками  (рестарт) и  (пуск).

Если требуется корректировка исходных текстов, ее можно

произвести как в этой же оболочке, открыв соответствующие файлы, так и в основной среде. После сохранения изменений на- жать кнопку  , подождать, пока не закончится компиля- ция и диаграмма не очистится, после чего запустить моделирова- ние ( ).

делены другим цветом). В скриншот включить: коммутационную матрицу, логический элемент, внутреннюю схему логического элемента.

###### Модификация проекта

Осуществить модификацию проекта по одному из следую- щих заданий, продемонстрировать работу, в распечатках исход- ных текстов отметить произведенную коррекцию.

* 1. При нажатии кнопки производить переключение не оди- ночных индикаторов, а пар (или групп) индикаторов.
	2. Добавить кнопку или переключатель, посредством кото- рых изменять направление переключения индикаторов.
	3. При постоянно нажатой кнопке осуществить автопе- реключение индикаторов с некоторым периодом (0.5…2 с).
	4. Вместо переключателей использовать дополнительную кнопку, при каждом нажатии на которую циклически изменять частоту мерцания индикаторов.

###### Реализация иерархического проекта

Ознакомиться с проектом:

##### D: \ PLIS \ CompDesign \ MusicSound \ MusicSound.xise

В проекте показаны: иерархическая структура (содержит не- сколько компонентов); реализация постоянной памяти (ROM) и работа с ней; обслуживание датчика вращения – энкодера – с устранением дребезга контактов.

Энкодер используется в качестве регулятора (задатчика ча- стоты), его кнопка также разрешает вывод сигнала на соедини- тель Audio jack, куда рекомендуется подключить звуковой излу- чатель (наушники).

Откомпилировать проект, загрузить в отладочный модуль, проверить работу. **Привести в отчете** схему соединений компо- нентов (RTL Schematic).

Дополнительные задания.

* 1. С помощью кнопок (или переключателей) изменять ха- рактер (уровень громкости) генерируемого тона путем увеличе- ния скважности импульсов.
	2. Сформировать сигнал с плавно возрастающей / убываю- щей частотой ("сирена").
	3. Реализовать выдачу короткого тонального сигнала с не- которой периодичностью (каждую минуту, секунду и т.п.).

## Содержание отчета

Функциональная схема аппаратного модуля, тексты про- грамм, результаты (перечислены в каждом пункте выполнения работы).

###### Просмотр технической документации

Перейти в режим *Implementation*. Двойным щелчком вызвать нижеперечисленные процедуры, **привести в отчете** скриншоты основных результатов.

* 1. **Design Summary/Reports**

Привести таблицы *Project Status*, *Device Utilization*. Из этого окна также можно вывести подробные протоколы и результаты всех этапов реализации.

##### Synthesize - XST: View RTL Schematic

Схема устройства на уровне описания цепей и регистров. Обратить внимание, что оптимизированная схема может не соот- ветствовать ее описанию в исходных файлах, например, в данном проекте демультиплексор заменен мультиплексорами.

##### Synthesize - XST: View Technology Schematic

Схема устройства на уровне стандартных логических эле- ментов, содержащихся в конкретной ПЛИС. Найти на этой схеме элемент с 3-4 входами (lut3, lut4), раскрыть его, просмотреть внутреннюю схему, логическую функцию, таблицу истинности, карту Карно.

##### Implement Design: Place & Route: View/Edit Routed De- sign (FPGA Editor)

Запустится отдельная программа со схемой логических ячеек на кристалле и трассировкой соединений. Подобрать необходи- мый масштаб, найти используемые элементы и линии связи (вы-

## Библиографический список

1. Максфилд К. Проектирование на ПЛИС. Курс молодого бойца. — М.: Издательский дом «Додэка-XXI», 2007. — 408 с.
2. Тарасов И. Е., Певцов Е.Ф. Программируемые логические схемы и их применение в схемотехнических решениях : Учеб. пособие. — М.: МИРЭА, 2012. — 181 с. — Режим доступа: [http://www.edamc.mirea.ru/files/Pevtsov&Tarasov.pdf](http://www.edamc.mirea.ru/files/Pevtsov%26Tarasov.pdf)
3. Поляков А.К. Языки VHDL и VERILOG в проектировании цифровой аппаратуры [Электронный ресурс] — М.: СОЛОН- Пресс, 2009. — 314 с. — Режим доступа: <http://e.lanbook.com/books/element.php?pl1_id=13656>

## Контрольные вопросы

1. Перечислите и охарактеризуйте основные этапы проекти- рования цифровых устройств на основе ПЛИС.
2. Какие элементы содержатся на кристалле ПЛИС?
3. Каковы правила описания входных и выходных портов устройства?
4. Охарактеризуйте основные операторы языка HDL.
5. Какая информация содержится в файле проектных ограни- чений?
6. Какова цель проведения моделирования? Чем отличаются проекты, исходные данные для реализации и для моделирования?
7. Какая документация может быть подготовлена компью- терной системой проектирования?

# Исходные тексты для проекта begproject

#### beg\_source.vhd

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

‐‐ УЧЕБНЫЙ ПРОЕКТ НАЧАЛЬНОГО УРОВНЯ

‐‐ Каждое нажатие одной из кнопок (BTN\_SOUTH) вызывает последовательную активизацию

‐‐ одного из 8 индикаторов, при этом данный индикатор мигает с частотой, которую

‐‐ можно выбирать переключателями SW0, SW1.

‐‐ В проекте синтезируются:

‐‐ делитель входной тактовой частоты 50 МГц ‐> 24 Гц;

‐‐ 3 делителя частоты на два для получения частот 12, 6, 3 Гц;

‐‐ 3‐х разрядный двоичный счетчик нажатий кнопки (счет: 0, 1, ..., 7, 0, ...);

‐‐ демультиплексор‐дешифратор, выбирающий по коду счетчика один из выходов на индикаторы;

‐‐ мультиплексор (коммутатор) сигналов с разными частотами на выход, соединенный

‐‐ с информационным входом демультиплексора

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

‐‐ Объявление библиотек

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL; use IEEE.STD\_LOGIC\_ARITH.ALL; use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

‐‐ Описание входных и выходных сигналов устройства

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

entity beg is

port ( CLK : in std\_logic; ‐‐ Вход тактовой частоты 50 MHz BTN\_SOUTH : in std\_logic; ‐‐ Сигнал от кнопки

SW : in std\_logic\_vector(1 downto 0); ‐‐ Сигналы от переключателей

LED : out std\_logic\_vector(7 downto 0) ‐‐ Выходы на индикаторы

);

end beg;

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

‐‐ Начало описания архитектуры устройства

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

architecture beg\_arch of beg is

‐‐‐‐‐‐‐ Объявление внутренних сигналов

‐‐ Сигналы разных частот мигания индикаторов

signal clk24, clk12, clk6, clk3 : std\_logic := '0';

‐‐ Счетчик‐делитель входной тактовой частоты (21‐разрядный) signal div\_clk : std\_logic\_vector(20 downto 0) := (others => '0');

‐‐ Циклический счетчик нажатий кнопки (работает на вычитание) signal push\_counter : std\_logic\_vector(2 downto 0) := "111";

‐‐ Предыдущее состояние кнопки

signal btn\_prev : std\_logic;

‐‐ Сигнал выбранной частоты (с выхода мультиплексора на вход демультиплексора) signal freq : std\_logic;

‐‐ Выходы демультиплексора (для непосредственного подключения к индикаторам) signal dmux\_out : std\_logic\_vector(7 downto 0);

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

‐‐ Описание функционирования устройства

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

begin

‐‐ Процесс генерации внутренней тактовой частоты 24 Гц,

‐‐ в качестве выхода используется самый старший разряд счетчика

div\_clk <= div\_clk + 1 when rising\_edge(CLK);

clk24 <= div\_clk(20); ‐‐ #### В режиме симуляции задать div\_clk(10) ####

‐‐ Процессы генерации остальных тактовых частот (12 Гц, 6 Гц, 3 Гц) clk12 <= not clk12 when rising\_edge(clk24);

clk6 <= not clk6 when rising\_edge(clk12); clk3 <= not clk3 when rising\_edge(clk6);

‐‐ Мультиплексор (переключатель) частот clk3, clk6, clk12, clk24 на один выход freq

‐‐ в зависимости от положения переключателей SW(0), SW(1) with SW select

freq <= clk3 when "00",

clk6 when "01",

clk12 when "10", clk24 when others;

‐‐ Счетчик нажатий кнопки (этот вариант не учитывает дребезг контактов)

process (BTN\_SOUTH) begin

‐‐ Процесс запускается при нажатии/отжатии кнопки

if rising\_edge(BTN\_SOUTH) then ‐‐ Условие: возрастающий фронт сигнала (т.е. нажатие) push\_counter <= push\_counter ‐ 1; ‐‐ Декремент счетчика

end if; end process;

‐‐ Счетчик нажатий кнопки (вариант с фильтрацией дребезга)

‐‐ Идентификация нажатия осуществляется по фронтам тактового сигнала (с периодом около 40 мс)

‐‐ и при условии, что новое состояние соответствует нажатию, а предыдущее ‐ отжатию.

‐‐process (clk24)

‐‐begin

‐‐ if rising\_edge(clk24) then ‐‐ Условие: возрастающий фронт сигнала

‐‐ if (BTN\_SOUTH = '1' and btn\_prev = '0') then

‐‐ push\_counter <= push\_counter ‐ 1;

‐‐ end if;

‐‐ btn\_prev <= BTN\_SOUTH; ‐‐ Запоминание текущего состояния в качестве предыдущего

‐‐ end if;

‐‐end process;

‐‐ Демультиплексор ‐ передает сигнал с выбранной частотой (freq) на один из 8 выходов

‐‐ dmux\_out в зависимости от состояния счетчика нажатий (push\_counter) process (push\_counter, freq)

begin

dmux\_out <= (others => '0'); ‐‐ На все выходы по умолчанию выводится лог. '0' case (push\_counter) is

when "000" => dmux\_out(0) <= freq; ‐‐ На выбранном активном выходе формируется сигнал

when "001" => dmux\_out(1) <= freq; ‐‐ с заданной частотой

when "010" => dmux\_out(2) <= freq; when "011" => dmux\_out(3) <= freq; when "100" => dmux\_out(4) <= freq; when "101" => dmux\_out(5) <= freq; when "110" => dmux\_out(6) <= freq; when "111" => dmux\_out(7) <= freq;

end case; end process;

‐‐ Подключение индикаторов

LED <= dmux\_out;

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

end beg\_arch; ‐‐ КОНЕЦ описания архитектуры и функционирования

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

#### s3ask.ucf

############################################################################## #

# ФАЙЛ ПРОЕКТНЫХ ОГРАНИЧЕНИЙ

# ##############################################################################

##############################################################################

# Входы тактовых частот 50 МГц (CLK) и 133 МГц (CLK\_AUX) ##############################################################################

NET "CLK" NET "CLK\_AUX"

LOC = "E12" | IOSTANDARD = LVCMOS33 | PERIOD = 20.000 ; LOC = "V12" | IOSTANDARD = LVCMOS33 | PERIOD = 7.5 ;

##############################################################################

# Дискретные светодиодные индикаторы (LED) ##############################################################################

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NET "LED<0>" | LOC = "R20" | | IOSTANDARD = LVCMOS33 | DRIVE = 8 |  | SLEW = | SLOW | ; |
| NET "LED<1>" | LOC = "T19" | | IOSTANDARD = LVCMOS33 | DRIVE = 8 |  | SLEW = | SLOW | ; |
| NET "LED<2>" | LOC = "U20" | | IOSTANDARD = LVCMOS33 | DRIVE = 8 |  | SLEW = | SLOW | ; |
| NET "LED<3>" | LOC = "U19" | | IOSTANDARD = LVCMOS33 | DRIVE = 8 |  | SLEW = | SLOW | ; |
| NET "LED<4>" | LOC = "V19" | | IOSTANDARD = LVCMOS33 | DRIVE = 8 |  | SLEW = | SLOW | ; |
| NET "LED<5>" | LOC = "V20" | | IOSTANDARD = LVCMOS33 | DRIVE = 8 |  | SLEW = | SLOW | ; |
| NET "LED<6>" | LOC = "Y22" | | IOSTANDARD = LVCMOS33 | DRIVE = 8 |  | SLEW = | SLOW | ; |
| NET "LED<7>" | LOC = "W21" | | IOSTANDARD = LVCMOS33 | DRIVE = 8 |  | SLEW = | SLOW | ; |

##############################################################################

# Символьный дисплей (LCD) ‐ 4‐разрядный интерфейс

##############################################################################

NET "LCD\_D<4>" NET "LCD\_D<5>" NET "LCD\_D<6>" NET "LCD\_D<7>" NET "LCD\_E" NET "LCD\_RS" NET "LCD\_RW"

LOC = "AA12" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ; LOC = "Y16" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ; LOC = "AB16" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ; LOC = "Y15" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ; LOC = "AB4" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ; LOC = "Y14" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ; LOC = "W13" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ;

##############################################################################

# Кнопки общего назначения ‐ при нажатии формируют на входе FPGA лог. '1'. # На входе обязательны подтягивающие резисторы к лог. '0'.

##############################################################################

NET "BTN\_WEST" NET "BTN\_SOUTH" NET "BTN\_EAST" NET "BTN\_NORTH"

LOC = "U15" | IOSTANDARD = LVCMOS33 | PULLDOWN ; #BTN2 LOC = "T15" | IOSTANDARD = LVCMOS33 | PULLDOWN ; #BTN3 LOC = "T16" | IOSTANDARD = LVCMOS33 | PULLDOWN ; #BTN4 LOC = "T14" | IOSTANDARD = LVCMOS33 | PULLDOWN ; #BTN5

NET "BTN\_SOUTH" CLOCK\_DEDICATED\_ROUTE = FALSE;

##############################################################################

# Механические переключатели (SW) ‐ формируют лог. '0' и '1'.

# Используется подтягивающий резистор к лог. '1' (не обязателен) ##############################################################################

|  |  |  |
| --- | --- | --- |
| NET "SW<0>" | LOC = "V8" | | IOSTANDARD = LVCMOS33 | PULLUP ; |
| NET "SW<1>" | LOC = "U10" | | IOSTANDARD = LVCMOS33 | PULLUP ; |
| NET "SW<2>" | LOC = "U8" | | IOSTANDARD = LVCMOS33 | PULLUP ; |
| NET "SW<3>" | LOC = "T9" | | IOSTANDARD = LVCMOS33 | PULLUP ; |

##############################################################################

# Энкодер (ROT) с двухфазным выходом (A, B) и кнопкой выбора (CENTER). # При нажатии кнопки формируется лог. '1' (требуется подтяжка к '0').

# Выходы A, B ‐ механические контакты на лог. '0' (требуется подтяжка к '1'). ##############################################################################

NET "ROT\_CENTER" NET "ROT\_A"

NET "ROT\_B"

LOC = "R13" | IOSTANDARD = LVCMOS33 | PULLDOWN ; LOC = "T13" | IOSTANDARD = LVCMOS33 | PULLUP ; LOC = "R14" | IOSTANDARD = LVCMOS33 | PULLUP ;

##############################################################################

# Стерео аудиовыход (AUD) ##############################################################################

NET "AUD\_L" NET "AUD\_R"

LOC = "Y10" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ; LOC = "V10" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ;

#### beg\_tb.vhd

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

‐‐ Company:

‐‐ Engineer:

‐‐

‐‐ Create Date:

‐‐ Design Name:

‐‐ Module Name: D:/PLIS/CompDesign/BegProject/beg\_tb.vhd

‐‐ Project Name: begproject

‐‐ Target Device:

‐‐ Tool versions:

‐‐ Description:

‐‐

‐‐ VHDL Test Bench Created by ISE for module: beg

‐‐

‐‐ Dependencies:

‐‐

‐‐ Revision:

‐‐ Revision 0.01 ‐ File Created

‐‐ Additional Comments:

‐‐

‐‐ Notes:

‐‐ This testbench has been automatically generated using types std\_logic and

‐‐ std\_logic\_vector for the ports of the unit under test. Xilinx recommends

‐‐ that these types always be used for the top‐level I/O of a design in order

‐‐ to guarantee that the testbench will bind correctly to the post‐implementation

‐‐ simulation model.

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL; USE ieee.std\_logic\_unsigned.ALL;

‐‐ Uncomment the following library declaration if using

‐‐ arithmetic functions with Signed or Unsigned values

‐‐USE ieee.numeric\_std.ALL;

ENTITY beg\_tb IS END beg\_tb;

ARCHITECTURE behavior OF beg\_tb IS

‐‐ Component Declaration for the Unit Under Test (UUT) COMPONENT beg

PORT(

CLK : IN std\_logic; BTN\_SOUTH : IN std\_logic;

SW : IN std\_logic\_vector(1 downto 0); LED : OUT std\_logic\_vector(7 downto 0)

);

END COMPONENT;

‐‐Inputs

signal CLK : std\_logic := '0'; signal BTN\_SOUTH : std\_logic := '0';

signal SW : std\_logic\_vector(1 downto 0) := (others => '0');

‐‐Outputs

signal LED : std\_logic\_vector(7 downto 0);

‐‐ Clock period definitions

constant CLK\_period : time := 20 ns; BEGIN

‐‐ Instantiate the Unit Under Test (UUT) uut: beg PORT MAP (

CLK => CLK,

BTN\_SOUTH => BTN\_SOUTH, SW => SW,

LED => LED

);

‐‐ Clock process definitions CLK\_process :process

begin

CLK <= '0';

wait for CLK\_period/2;

CLK <= '1';

wait for CLK\_period/2; end process;

‐‐ Stimulus process stim\_proc: process begin

‐‐ Реализация последовательности формирования входных сигналов

‐‐ В режиме симуляции временные дискреты us (микросекунды)

‐‐ соответствуют реальным ms (миллисекундам)

SW <= "11";

wait for 1000 us;

‐‐ Начальное состояние

‐‐ Задержка

‐‐ Цикл

for i in 1 to 12 loop

BTN\_SOUTH <= '1'; ‐‐ Имитация нажатия кнопки

wait for 100 us; ‐‐ (на 100 мс)

BTN\_SOUTH <= '0'; ‐‐ Имитация отжатия кнопки

wait for 900 us; SW <= SW ‐ 1;

‐‐ (на 900 мс)

‐‐ Имитация изменения состояния переключателей

‐‐ ( "11" ‐> "10" ‐> "01" ‐> "00" ‐> "11 ‐> ...)

BTN\_SOUTH <= '1'; ‐‐ Имитация нажатия кнопки

wait for 600 us; ‐‐ (на 600 мс)

BTN\_SOUTH <= '0'; ‐‐ Имитация отжатия кнопки

wait for 400 us; ‐‐ (на 400 мс) end loop;

wait; ‐‐ Остановка

end process; END;