5321

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

КОНСТРУИРОВАНИЕ МИКРО- И НАНОСИСТЕМ

Методические указания
к лабораторной работе № 2

Проектирование топологии тонкопленочного конденсатора

Рязань 2018

УДК 621.315.592

Конструирование микро- и наносистем: методические указания к лабораторной работе № 2 / Рязан. гос. радиотехн. ун-т; сост.: Н.В. Рыбина, Н.Б. Рыбин., Д.С. Кусакин. Рязань, 2018. 16 с.

Содержат краткие теоретические сведения о конструкциях и способах расчета топологии тонкопленочных конденсаторов для гибридных интегральных микросхем, а также материалы для выполнения лабораторной работы по дисциплине “Конструирование микро- и наносистем”.

Предназначены для студентов дневного отделения направления 11.03.04.

Ил. 2. Табл. 2. Библиогр.: 2 назв.

Тонкопленочный конденсатор, гибридная интегральная схема, проектирование

Печатается по решению редакционно-издательского совета Рязанского государственного радиотехнического университета.

Рецензент: кафедра микро- и наноэлектроники РГРТУ (зав. кафедрой Т.А. Холомина)

Конструирование микро- и наносистем

Составители: Р ы б и н а Наталья Владимировна
Р ы б и н Николай Борисович
К у с а к и н Дмитрий Сергеевич

Редактор Р.К. Мангутова
Корректор М.Е. Цветкова
Подписано в печать 10.10.18. Формат бумаги 60×84 1/16.
Бумага писчая. Печать трафаретная. Усл. печ. л. 1,0.
Тираж 30 экз. Заказ
Рязанский государственный радиотехнический университет.
390005, Рязань, ул. Гагарина, 59/1.
Редакционно-издательский центр РГРТУ.

Цель работы

Изучение порядка расчета тонкопленочных конденсаторов в гибридных интегральных микросхемах.

1. Краткие теоретические сведения

Интегральная микросхема (ИМС) – это конструктивно законченное изделие электронной техники, выполняющее определенную функцию преобразования информации и содержащее совокупность электрически связанных между собой электрорадиоэлементов (ЭРЭ), изготовленных в едином технологическом цикле [1].

Термин «интегральная микросхема» отражает: объединение значительного числа транзисторов, диодов, конденсаторов, резисторов и соединяющих проводников в единую конструкцию (конструктивная интеграция); выполнение схемой функций преобразования информации, более сложных по сравнению с функциями отдельных ЭРЭ (схемотехническая интеграция); выполнение в едином технологическом цикле одновременно всех ЭРЭ схемы и межсоединений и одновременное формирование групповым методом большого числа одинаковых ИМС (технологическая интеграция).

По способу изготовления различают полупроводниковые и пленочные ИМС. В полупроводниковых ИМС все ЭРЭ и часть межсоединений сформированы в приповерхностном слое полупроводниковой (обычно кремниевой) подложки. В пленочных ИМС пассивные ЭРЭ изготовлены в виде совокупности тонких (менее 1 мкм) или толстых (10 – 50 мкм) пленок, нанесенных на диэлектрическую подложку. Гибридные ИМС (ГИМС) представляют

собой комбинацию пленочных пассивных ЭРЭ с миниатюрными бескорпусными дискретными активными приборами (полупроводниковыми ИМС, транзисторами, диодами), расположенных на общей диэлектрической подложке. ЭРЭ, которые являются неотъемлемой составной частью ИМС и не могут быть выделены из нее как самостоятельное изделие, называют элементами ИМС, а дискретные активные ЭРЭ ГИМС – навесными компонентами (или просто компонентами), подчеркивая тем самым, что их изготавливают отдельно в виде самостоятельных приборов, которые могут быть приобретены изготовителем ГИС как покупные изделия. В отличие от дискретных компонентов элементы ИМС называют интегральными (интегральный резистор, интегральный диод) [1].

В совмещенных ИМС ЭРЭ выполнены в приповерхностном слое полупроводникового кристалла (как у полупроводниковой ИМС), а пассивные нанесены в виде пленок на покрытую диэлектриком поверхность того же кристалла (как у пленочной ИМС).

ГИС по сравнению с полупроводниковыми имеют ряд преимуществ: обеспечивают широкий диапазон номиналов, меньшие пределы допусков и лучшие электрические характеристики пассивных элементов (более высокую добротность, температурную и временную стабильность, меньшее число и менее заметное влияние паразитных элементов).

Одна из основных характеристик микроэлектронного изделия – рассеиваемая мощность. При гибридном исполнении можно обеспечить изготовление изделий достаточно большой мощности, что важно при создании аналоговых устройств, управляющих мощными входными цепями.

Конструкции пленочных конденсаторов

Так как емкость тонкопленочного конденсатора определяется площадью перекрытия его обкладок (активной площадью или площадью верхней обкладки), то он может быть изготовлен в нескольких конструктивных вариантах, приведенных на рис. 1 [2].

На рис. 1, а приведена конструкция конденсатора с площадью верхней обкладки более 5 мм^2. Так как верхние обкладки формируют масочным методом, то для устранения погрешности совмещения маски в месте вывода верхней обкладки с противоположной стороны от вывода делают компенсатор. При значительной площади верхней обкладки эта погрешность мала и компенсатор не нужен. При активной площади от 1 до 5 мм^2 обкладки конденсатора можно выполнять в виде двух взаимно пересекающихся полосок (рис. 1, б). Если расчетная площадь конденсатора меньше 1 мм^2, конденсатор можно выполнять в виде последовательно соединенных конденсаторов (рис. 1, в). Если расчетная площадь слишком мала и не позволяет конструировать конденсатор приемлемых размеров, можно использовать в качестве диэлектрика подложку (рис. 1, г), которая должна быть пригодна для напыления обкладок с обеих сторон. Можно конструировать также гребенчатый конденсатор (рис. 1, д). Емкость такого конденсатора почти целиком определяется емкостью, обусловленной краевым эффектом.

Материалы конденсаторов. Обкладки конденсаторов должны иметь высокую проводимость, коррозионную стойкость, технологическую совместимость с материалом подложки и диэлектрика конденсатора: температурные коэффициенты линейного расширения (ТКЛР), близкие к ТКЛР подложки и диэлектрика, хорошую адгезию к подложке и диэлектрику, высокую механическую прочность.

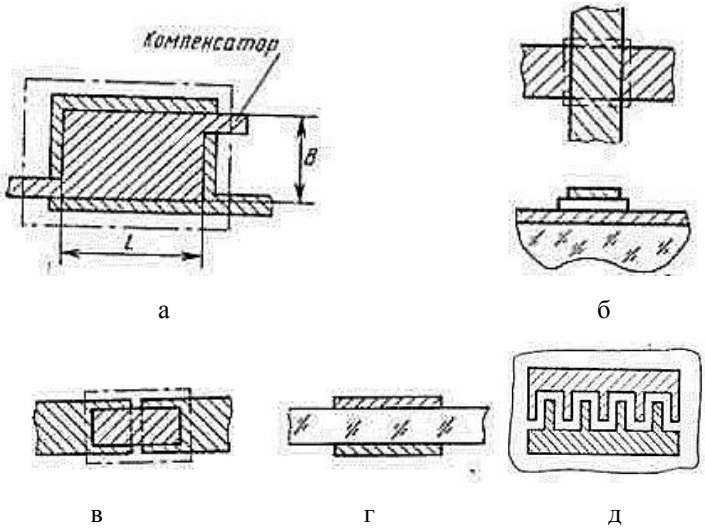


Рис. 1. Конфигурации пленочных конденсаторов: а – площадь верхней обкладки более 5 мм^2; б – площадь от 1 до 5 мм^2; в – расчетная площадь конденсатора меньше 1 мм^2; г – конструкция с использованием подложки в качестве диэлектрика; д – гребенчатый конденсатор [1]

Наилучшим материалом для обкладок конденсаторов является алюминий, который, однако, имеет плохую адгезию к подложке. Для предотвращения отслаивания нижней обкладки вначале напыляют подслой титана или ванадия. Верхняя обкладка, напыляемая на диэлектрик, не требует подслоя. Применение золота для обкладок не рекомендуется из-за высокой подвижности атомов и возможной диффузии сквозь диэлектрик, приводящей к короткому замыканию обкладок.

Материал диэлектрика должен иметь хорошую адгезию к подложке и материалу обкладок, обладать высокой электрической

прочностью и малыми потерями, иметь высокую диэлектрическую проницаемость и минимальную гигроскопичность, не разлагаться в процессе формирования пленок. В качестве диэлектрических материалов наиболее часто используют моноокиси кремния и германия. В табл. 1 приведены основные параметры диэлектрических материалов тонкопленочных конденсаторов.

Конструкции пленочных конденсаторов должны учитывать особенности топологической структуры пленочного функционального узла (размеры подложки, количество и расположение выводов и т.д.), величину номинала, характеристики используемых материалов, технологию производства, требуемую и возможную точность воспроизведения номиналов, условия эксплуатации микросхем [2]. В соответствии с этим основными исходными данными для проектирования резисторов являются:

- номинальная емкость C_i, Ом;
- погрешность номинальной емкости γ_{Ci}, %;
- рабочий диапазон температур, $T_{\max} - T_{\min}$, ºС;
- технологические ограничения;
- шаг координатной сетки, мм;
- напряжение питания схемы $U_{раб}$, В;
- диапазон рабочих частот.

Технологические ограничения при использовании различных методов создания тонкопленочных конденсаторов [масочного (М), фотолитографического (Ф), комбинированного масочного и фотолитографического (МФ), электронно-ионного (ЭИ) и по танталовой технологии (ТА)] приведены в табл. 2 [1].

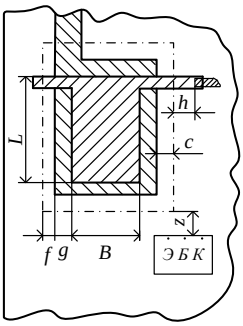
Таблица 1. Параметры диэлектрических материалов тонкопленочных конденсаторов [1]

Материал		Параметры							
диэлектрика	обкладок	Удельное поверхностное сопротивление обкладок ρs, Ом/□	Удельная емкость C_0, пФ/см^2	Рабочее напряжение $U_{раб}$, В	Диэлектрическая проницаемость ε при f = 1 кГц	Тангенс угла диэлектрических потерь tgδ при f = 1 кГц	Электрическая прочность $E_{пр}$, В/см	Рабочая частота f, МГц	Температурный коэффициент емкости ТКС при T=-60÷125 °С
Моноокись кремния	Алюминий А99	0,2	5000	60	5,0-6,0	0,01-0,02	$(2\text{-}3)\cdot10^6$	500	$2\cdot10^{-4}$
			10000	30					
Моноокись германия			5000	10	11-12	0,005-0,007	$1{,}0\cdot10^6$	300	$3\cdot10^{-4}$
			10000	7					
			15000	5					
Боросили-катное стекло			2500	24	4	0,001-0,0015	$(3\text{-}4)\cdot10^6$	300	$0{,}35\cdot10^{-4}$
			5000	15					
			10000	10					
			15000	8					

Окончание таблицы 1

7	Стекло электроваку-умное			15000	12,6	5,2	0,002-0,003	$(3\text{-}4)\cdot10^{6}$	300	$(0{,}5\text{-}1)\cdot10^{-4}$ при T=-60÷25 °C, $(1{,}5\text{-}1{,}8)\cdot10^{-4}$ при T=25÷155 °C
				20000	10-12,6					
				30000	6,3-10					
				40000	6,3					
	Пятиокись тантала	Тантал ТВЧ, нижняя обкладка	1-10	60000	15	23	0,02	$2\cdot10^{6}$	0,1	$4\cdot10^{-4}$
				100000	10					
		Алюминий А99, верхняя обкладка	0,2	200000	3					

Таблица 2. Конструкторско-технологические ограничения при проектировании тонкопленочных конденсаторов [1, 2]

Топология конденсатора	Содержание ограничения	Размер ограничения, мм, при использовании метода				
		М	Ф	МФ	ЭИ	ТА
	Точность изготовления линейных размеров элемента Δl, Δb, Δa, ΔL, ΔB, мм	±0,01	±0,01	±0,01	±0,01	±0,01
	Минимально допустимые расстояния, мм: - между краями диэлектрика и нижней обкладкой конденсатора f	0,1	0,1	0,1	0,1	-
	Между краями верхней и нижней обкладок g	0,2				
	Между краем диэлектрика и нижней обкладкой в месте вывода верхней обкладки c	0,2				
	Между краем диэлектрика и соединением вывода конденсатора с другим пленочным элементом h	0,3				

Окончание таблицы 2

	От пленочного конденсатора до навесных компонентов *z*	0.5
	Минимальная площадь перекрытия обкладок *L*×*B*, мм	0,5×0,5
	Максимальное отклонение емкости конденсатора от номинального значения, %	±12

Порядок расчета тонкопленочных конденсаторов

Конденсатор показан на рис. 2.

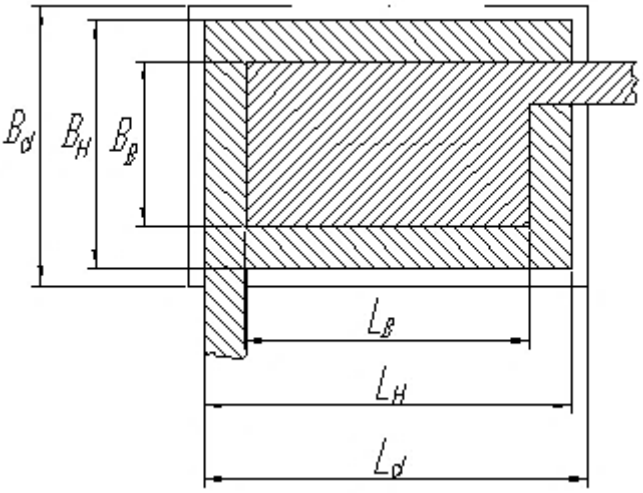


Рис. 2. Тонкопленочные конденсаторы с емкостью до $5 \cdot 10^5$ пФ

Общий порядок расчета (рис. 2) включает несколько приведенных ниже этапов.

1. Выбор материала диэлектрика по рабочему напряжению производится в соответствии с данными, приведенными в табл. 1. Следует ориентироваться на материал диэлектрика с возможно более высокими диэлектрической проницаемостью ε, электрической прочностью $E_{пр}$, малыми значениями *ТКС* и $tg\delta$ для обеспечения минимальных геометрических размеров конденсатора, тип которого приведен на рис. 2.

2. Определение минимальной толщины диэлектрика производится из условий обеспечения необходимой электрической прочности по формуле

$$d_{min} \geq k_з U_{раб} / E_{пр}, \qquad (1)$$

где $k_з$ – коэффициент запаса электрической прочности (обычно для тонкопленочных конденсаторов $k_з$ = 2...3); $E_{пр}$ – электрическая прочность материала диэлектрика, *В/мм*; $U_{раб}$ - рабочее напряжение. Толщину диэлектрика чаще выбирают в пределах 0,1...1 *мкм*. Минимальная толщина диэлектрического слоя ограничена требованием получения сплошной пленки без сквозных отверстий и с заданной электрической прочностью, а максимальная толщина ограничена механическими напряжениями в растущей пленке. Оптимальной толщиной принято считать величину в 0,3...0,5 *мкм*.

3. Определение удельной емкости конденсатора ($нФ/см^2$) ведется по формуле

$$C_{0v} = 0{,}0885\varepsilon / d, \qquad (2)$$

где d – толщина диэлектрика, *см*.

4. Определение относительной температурной погрешности производится из выражения

$$\gamma_{Ct} = \alpha_C \cdot (T_{max} - 20\ ^0\text{C}), \quad (3)$$

где α_C – температурный коэффициент емкости материала диэлектрика.

5. Определение допустимой относительной погрешности активной площади конденсатора осуществляется по формуле

$$\gamma_{S\ \text{доп}} = \gamma_C - \gamma_{Co} - \gamma_{\text{ст}} - \gamma_{Ct}, \quad (4)$$

где γ_{Co} – относительная погрешность удельной емкости, характеризующая воспроизводимость удельной емкости в условиях данного производства (зависит от материала и погрешности толщины диэлектрика и составляет 3 – 5 %); $\gamma_{ст}$ – относительная погрешность, обусловленная старением пленок конденсатора (зависит от материала и метода защиты и обычно не превышает 2 – 3 %).

6. Определение удельной емкости конденсатора с учетом точности изготовления:

$$C_{0\text{точн}} = C \cdot \left[\frac{\gamma_{S\text{доп}}}{2\Delta L}\right]^2 \frac{\text{К}_\text{ф}}{(1+\text{К}_\text{ф})^2}, \quad (5)$$

для обкладок квадратной формы (при $K_\phi = 1$)

$$C_{0\text{точн}} = C \cdot \left[\frac{\gamma_{S\text{доп}}}{2\Delta L}\right]^2. \quad (6)$$

Точность изготовления линейных размеров пленочных элементов и расстояний между ними Δl, Δb, ΔL, ΔB выбирается с учетом метода формирования элемента (табл. 2).

Определяем, какой должна быть удельная емкость конденсатора с учетом технологических возможностей изготовления по площади перекрытия обкладок и толщине диэлектрика. Задаемся S_{min} = 1 мм^2.

$$C_{0min} = \frac{C_{min}}{S_{min}}. \quad (7)$$

7. Выбор минимального значения удельной емкости конденсатора с учетом электрической прочности и точности изготовления

$$C_0 < min\ \{C_{0v};\ C_{0точн};\ C_{0min}\}.$$

Определяем, какая толщина диэлектрика соответствует выбранной удельной емкости C_0: $d = 0{,}0885\varepsilon / C_0$.

Далее проводим расчет геометрических размеров конденсатора.

8. Определение коэффициента, учитывающего краевой эффект:

$$K = \begin{cases} 1;\ C/C_0 \geq 5\ мм^2, \\ (1{,}3...0{,}06)\cdot C/C_0;\ 1 \leq C/C_0 \leq 5\ мм^2. \end{cases}$$

9. Определение площади верхней обкладки конденсатора:

$$S = \frac{C}{C_0 \cdot K}. \quad (8)$$

10. Определение размеров верхней обкладки конденсатора при $K_ф$ = 1:

$$L = B = \sqrt{S}. \quad (9)$$

11. Вычисление размеров нижней обкладки конденсатора с учетом допусков на перекрытие:

$$L_H = B_H = L + 2g, \quad (10)$$

где g – размер перекрытия нижней и верхней обкладок конденсатора (табл. 2).

12. Определение размеров диэлектрика:

$$L_Д = B_Д = L_H + 2f, \quad (11)$$

где f – размер перекрытия нижней обкладки и диэлектрика.

13. Определение площади, занимаемой конденсатором:

$$S_Д = L_Д \cdot B_Д. \quad (12)$$

Форма обкладок конденсатора выбирается на этапе разработки эскиза топологии. Сначала проектируется обкладка конденсатора квадратной формы, а при отсутствии места на чертеже топологии для расположения квадрата следует задаться одной из сторон конденсатора, коэффициентом формы обкладок и вычислить размеры обкладок прямоугольной формы.

Для контроля емкости в процессе или после изготовления микросхемы конденсатор проектируется со специальными контактными площадками.

14. Конденсатор спроектирован правильно при выполнении приведенных ниже условий:

а) рабочий тангенс угла диэлектрических потерь не превышает заданного:

$$tg\delta_{раб} \leq tg\delta,$$

где $tg\delta_{раб} = tg\delta_Д + tg\delta_{об}$.

Тангенс угла потерь в обкладках $tg\delta_{об}$ следует вычислять по формуле

$$tg\delta_{об} \approx (4\pi / 3) \cdot f_{max} \cdot R_{об} \cdot C, \quad (13)$$

где $R_{об}$ – сопротивление обкладок конденсатора, *Ом*; C – емкость конденсатора, *Ф*; f_{max} – максимальная рабочая частота, *Гц*.

Сопротивление обкладок конденсатора зависит от его формы и рассчитывается по формуле

$$R_{об} = \rho_{Sоб} \cdot K_{ф}, \quad (14)$$

где $\rho_{Sоб}$ – удельное поверхностное сопротивление материала обкладок;

б) рабочая напряженность электрического поля $E_{раб}$ не превышает $E_{пр}$ материала диэлектрика $E_{раб} \leq E_{пр}$:

$$\mathrm{E_{раб}} = \frac{U_{\mathrm{раб}}}{d}; \quad (15)$$

в) погрешность активной площади конденсатора не превышает допустимую:

$$\gamma_{S\,\mathrm{раб}} \leq \gamma_{S\,\mathrm{доп}},$$

где $\gamma_{S\,\mathrm{раб}}$ вычисляется по формуле

$$\gamma_{S\mathrm{раб}} = \Delta L \cdot \frac{1+K_{\mathrm{ф}}}{\sqrt{K_{\mathrm{ф}} \cdot S}}. \quad (16)$$

Проверки показывают, что конденсатор не выходит за пределы точности, имеет запас по электрической прочности и активной площади, тангенс угла диэлектрических потерь меньше заданного.

2. Последовательность выполнения работы

1. Изучите теоретическую часть работы.

2. Получите задание в виде принципиальной схемы у преподавателя.

3. Выберите материалы диэлектрика и обкладок конденсатора.

4. Рассчитайте конструкции всех конденсаторов.

5. Сделайте проверку расчетов.

6. Определите суммарную площадь, занимаемую всеми конденсаторами.

3. Содержание отчета

1. Цель работы.

2. Результаты расчетов.

3. Чертежи тонкопленочных конденсаторов в масштабе с указанием размеров.

4. Выводы по полученным результатам.

Контрольные вопросы

1. Что такое гибридная интегральная схема?

2. Назовите требования, предъявляемые к материалам диэлектриков конденсаторов.

3. Назовите требования, предъявляемые к материалам обкладок конденсаторов.

4. Назовите основные материалы пленок диэлектриков конденсаторов ГИМС.

5. Назовите основные материалы обкладок конденсаторов ГИМС.

6. Какие конструкции бывают у тонкопленочных конденсаторов?

7. Что такое краевой эффект в конденсаторе?

8. Что такое коэффициент формы тонкопленочного резистора?

9. Как работает гребенчатый конденсатор?

10. Конструкции конденсаторов малой емкости.

11. Порядок расчета конденсатора.

12. Расскажите о подстраиваемых конденсаторах.

13. Как проводится проверка правильности расчета конструкции конденсатора?

Библиографический список

1. Коледов Л.А. Конструирование и технология микросхем: учеб. пособие для вузов. М.: Высшая школа, 1984. 232 с.

2. Меркулов А.И., Меркулов В.А. Основы конструирования интегральных микросхем. Самара: Издательство СГАУ, 2013. 270 с.

5200

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

КОНСТРУИРОВАНИЕ МИКРО- И НАНОСИСТЕМ

Методические указания
к лабораторной работе № 1

Проектирование топологии тонкопленочных резисторов

Рязань 2017

УДК 621.315.592

Конструирование микро- и наносистем: методические указания к лабораторной работе № 1 / Рязан. гос. радиотехн. ун-т; сост.: Н.В. Рыбина, Н.Б. Рыбин. Рязань, 2017. 16 с.

Содержат краткие теоретические сведения о конструкциях и способах расчета топологии тонкопленочных резисторов для гибридных интегральных микросхем, а также материалы для выполнения лабораторной работы по дисциплине “Конструирование микро- и наносистем”.

Предназначены для студентов дневного отделения направления 11.03.04

Ил. 3. Табл. 2. Библиогр.: 2 назв.

Тонкопленочный резистор, гибридная интегральная схема, проектирование

Печатается по решению редакционно-издательского совета Рязанского государственного радиотехнического университета.

Рецензент: кафедра микро- и наноэлектроники (зав. кафедрой Т.А. Холомина)

Конструирование микро- и наносистем

Составители: Р ы б и н а Наталья Владимировна
Р ы б и н Николай Борисович

Редактор Р.К. Мангутова
Корректор С.В. Макушина
Подписано в печать 25.09.17. Формат бумаги 60×84 1/16.
Бумага писчая. Печать трафаретная. Усл. печ. л. 1,0.
Тираж 30 экз. Заказ
Рязанский государственный радиотехнический университет.
390005, Рязань, ул. Гагарина, 59/1.
Редакционно-издательский центр РГРТУ.

Цель работы

Изучение порядка расчета тонкопленочных резисторов в гибридных интегральных микросхемах.

1. Краткие теоретические сведения

Интегральная микросхема (ИМС) – это конструктивно законченное изделие электронной техники, выполняющее определенную функцию преобразования информации и содержащее совокупность электрически связанных между собой электрорадиоэлементов (ЭРЭ), изготовленных в едином технологическом цикле [1].

Термин «интегральная микросхема» отражает: объединение значительного числа транзисторов, диодов, конденсаторов, резисторов и соединяющих проводников в единую конструкцию (конструктивная интеграция); выполнение схемой функций преобразования информации, более сложных по сравнению с функциями отдельных ЭРЭ (схемотехническая интеграция); выполнение в едином технологическом цикле одновременно всех ЭРЭ схемы и межсоединений и одновременное формирование групповым методом большого числа одинаковых ИМС (технологическая интеграция).

По способу изготовления различают полупроводниковые и пленочные ИМС. В полупроводниковых ИМС все ЭРЭ и часть межсоединений сформированы в приповерхностном слое полупроводниковой (обычно кремниевой) подложки. В пленочных ИМС пассивные ЭРЭ изготовлены в виде совокупности тонких (менее 1 мкм) или толстых (10 – 50 мкм) пленок, нанесенных на диэлектрическую подложку. Гибридные ИМС (ГИМС) представляют собой комбинацию пленочных пассивных ЭРЭ с миниатюрными бескорпусными дискретными активными приборами (полупроводниковыми ИМС, транзисторами, диодами), расположенных на общей диэлектрической подложке. ЭРЭ, которые являются неотъемлемой составной частью ИМС и не могут быть выделены из нее как самостоятельное изделие, называют элементами ИМС, а дискретные активные ЭРЭ ГИМС – навесными компонентами (или просто компонентами), подчеркивая тем самым, что их изготавливают отдельно в виде самостоятельных приборов, которые

могут быть приобретены изготовителем ГИС как покупные изделия. В отличие от дискретных компонентов элементы ИМС называют интегральными (интегральный резистор, интегральный диод) [1].

В совмещенных ИМС ЭРЭ выполнены в приповерхностном слое полупроводникового кристалла (как у полупроводниковой ИМС), а пассивные нанесены в виде пленок на покрытую диэлектриком поверхность того же кристалла (как у пленочной ИМС).

ГИС по сравнению с полупроводниковыми имеют ряд преимуществ: обеспечивают широкий диапазон номиналов, меньшие пределы допусков и лучшие электрические характеристики пассивных элементов (более высокая добротность, температурная и временная стабильность, меньшее число и менее заметное влияние паразитных элементов).

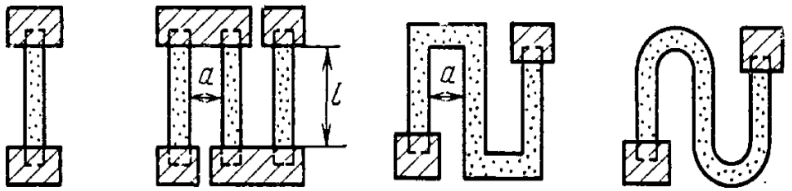
Одна из основных характеристик микроэлектронного изделия – рассеиваемая мощность. При гибридном исполнении можно обеспечить изготовление изделий достаточно большой мощности, что важно при создании аналоговых устройств, управляющих мощными входными цепями.

Конструкции пленочных резисторов

Резисторы являются наиболее распространенными элементами пленочных микросхем.

Параметры тонкопленочных резисторов определяются свойствами применяемых резистивных материалов (табл. 1). В качестве резистивных материалов используют чистые металлы, сплавы, а также специальные резистивные материалы (керметы). Для создания резисторов необходимы пленки с удельным поверхностным сопротивлением ρ_S от десятков до десятков тысяч Ом/□. Чем меньше толщина пленок, тем выше ρ_S, но одновременно повышается температурный коэффициент сопротивления α_R (TKR), а также ухудшается температурная и временная стабильность пленок.

Резистивные элементы гибридных микросхем получают напылением на диэлектрические основания (подложки) узких резистивных пленочных полосок различной конфигурации и контактных площадок, имеющих с ними некоторую зону перекрытия (рис. 1) [2].



а б в г

Рис. 1. Конфигурации тонкопленочных резисторов: а – полоска; б – составной из полосок; в – меандр; г – змейка [1]

Конструкции пленочных резисторов должны учитывать особенности топологической структуры пленочного функционального узла (размеры подложки, количество и расположение выводов и т.д.), величину номинала, характеристики используемых материалов, технологию производства, требуемую и возможную точность воспроизведения номиналов, условия эксплуатации микросхем [2]. В соответствии с этим основными исходными данными для проектирования резисторов являются:

- номинальное сопротивление R_i, Ом;
- погрешность номинального сопротивления γ_{Ri}, %;
- рабочий диапазон температур, $T_{\max} - T_{\min}$, ºС;
- технологические ограничения;
- шаг координатной сетки, мм;
- напряжение питания схемы E, В.

Технологические ограничения при использовании различных методов создания тонкопленочных резисторов [масочного (М), фотолитографического (Ф), комбинированного масочного и фотолитографического (МФ), электронно-ионного (ЭИ) и по танталовой технологии (ТА)] приведены в табл. 2 [1].

Таблица 1. Электрофизические характеристики материалов для плёночных резисторов и проводников [1, 2]

Материал	ρ_S, Ом/□	P_0, Вт/см^2	$\alpha_R \cdot 10^{-4}$, 1/град -60...+125, °С	Материал контактных площадок	Старение, % за 5000 ч в норм. усл.
Хром ХО, ГОСТ 5905-67	200... 500	2,0	0,6	Медь (луженая)	2,0
Нихром Х20Н80, ГОСТ12766-67	100...300	1,0	1	Медь	0,1...0,4
Сплав МЛТ-3М	300…500	2,0	2± ±(1,9...2,4)	Медь с подслоем ванадия (луженая); медь с подслоем нихрома (защищенная никелем)	±0,1
Кермет, ЕТО.021.033. ТУ	3000 5000 10000	2,0	3 -4 -5	Золото с подслоем хрома (нихрома)	±0,3
Тантал, СУО.021.041 ТУ	300...500	3,0	-2	Алюминий с подслоем ванадия; медь с подслоем нихрома; тантал	2,6-3,2
Сплавы: РС-4800 РС-3710 РС-3001 РС-1004, ЕТО.021.048ТУ	 $(0,3...1)10^3$ $(0,5...3)10^3$ $(0,8...1,5)10^3$ $(1...4)10^4$	2,0	 2,0 0,5...2 1,0 10,0	Золото с подслоем хрома (нихрома)	 2,0 2,0 1,0 2,0

Таблица 2. Конструкторско-технологические ограничения при проектировании тонкопленочных резисторов [1, 2]

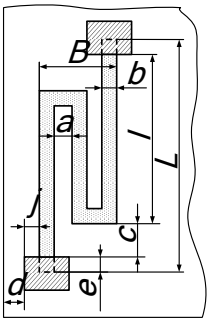
Топология резистора	Содержание ограничения		Размер ограничения, мм, при использовании метода				
			М	Ф	МФ	ЭИ	ТА
	Точность изготовления линейных размеров элемента и расстояний между ними Δl, Δb, Δa, ΔL, ΔB и других при расположении пленочных элементов в одном слое, мм		±0,01	±0,01	±0,01	±0,01	±0,01
	Минимально допустимый размер резистора, мм	*b*	0,1	0,1	0,1	0,15	0,05
		l	0,3	0,1	0,3	0,3	0,1
	Минимально допустимые расстояния между пленочными элементами, расположенными в одном слое, *a*, мм		0,3	0,1	0,3	0,1	0,05
	Максимально допустимое соотношение размеров *l*/*a*		10	100	30	100	100

Таблица 2. Окончание

	Минимально допустимое расстояние между пленочными элементами, расположенными в разных слоях, *c*, мм	0,2	0,1	0,2	0,1	0,1
	Перекрытия для совмещения пленочных элементов, расположенных в разных слоях, *e*, мм	≥0,2	≥0,1	≥0,2	≥0,1	≥0,1
	Минимальное расстояние от пленочных элементов до края платы *d*, мм	0,5	0,2	0,5	0,4	0,2
	Минимальная ширина пленочных проводников *i*, мм	0,1	0,05	0,1	0,1	0,05
	Минимально допустимое расстояние между краем пленочного резистора и краем его контактной площадки *j*, мм	0,2	0,1	0,2	0,1	0,1

Порядок расчета тонкопленочных резисторов

1. Конструктивный расчет тонкопленочных резисторов заключается в определении формы, геометрических размеров и минимальной площади, занимаемой резисторами на подложке. При этом необходимо, чтобы резисторы обеспечивали рассеивание

заданной мощности при удовлетворении требуемой точности в условиях существующих технологических возможностей.

Проектирование конструкции тонкопленочного резистора начинается с расчета мощности P_i, выделяемой на каждом резисторе. Грубо P_i можно оценить по формуле

$$P_i = \frac{E^2}{R_i}. \quad (1)$$

Однако в этом случае рассчитанное значение мощности получается завышенным, так как считается, что все напряжение питания падает на *i*-м резисторе. В итоге размеры резистора тоже могут получиться завышенными.

Более точно мощность можно рассчитать по формуле

$$P_i = I_i^2 \cdot R_i, \quad (2)$$

где I_i – ток, протекающий через *i*-й резистор. I_i рассчитывается исходя из заданной принципиальной схемы. Например, на рис. 2 приведен участок схемы, для которого требуется рассчитать мощность, рассеиваемую резисторами.

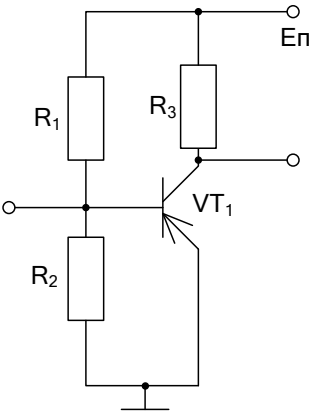


Рис. 2. Участок принципиальной схемы

Рассчитаем максимальную мощность, выделяемую на резисторе R_1. Предполагаем, что транзистор VT_1 в активном режиме (эмиттерный переход открыт, коллекторный – закрыт), тогда ток потечет от источника питания через резистор R_1 и эмиттерный переход на землю:

$$I_{R1} = E\text{п} / R_1, \quad (3)$$

следовательно, мощность будет равна

$$P_{R1} = I_{R1}^2 / R_1. \quad (4)$$

На резисторе R_2 максимальная мощность будет выделяться при открытом коллекторном и закрытом эмиттерном переходе транзистора VT_1. То есть ток, протекающий через резистор R_2, будет равен

$$I_{R2} = \frac{E_\text{п}}{R_2 + \frac{R_1 \cdot R_3}{R_1 + R_3}}, \quad (5)$$

следовательно, мощность будет равна

$$P_{R2} = I_{R2}^2 / R_2. \quad (6)$$

Аналогично рассчитывают все остальные мощности.

2. Особенностью расчета пленочных резисторов является групповое проектирование, что связано с необходимостью выбора резистивного материала. По возможности он должен быть единым для всех резисторов ИМС. Поэтому для пленочных ИМС при проектировании резисторов необходимо провести анализ номиналов резисторов, свести их в несколько групп с близкими значениями сопротивления, а затем уже выбрать для каждой группы материал.

Определяется оптимальное значение удельного поверхностного сопротивления резистивной пленки с точки зрения обеспечения минимума площади под резисторами гибридной ИМС по формуле

$$\rho_s = \sqrt{\frac{\sum_1^n R_i}{\sum_1^n \frac{1}{R_i}}}, \qquad (7)$$

где n – число резисторов; R_i – номинал i-го резистора.

3. Выбирается материал резистивной пленки (табл. 1) с удельным сопротивлением, ближайший по значению к вычисленному $\rho_{s\ опт}$. Используемые материалы регламентируются соответствующими ГОСТами и ОСТами. Следует обратить внимание на то, чтобы температурный коэффициент сопротивления (ТКС) был минимальным, а допустимая удельная мощность рассеивания P_0 была максимальной.

4. Проводится проверка правильности выбранного материала резистивной пленки с точки зрения точности изготовления резисторов.

Полная относительная погрешность изготовления пленочного резистора $\gamma_R = \Delta R/R$ состоит из суммы погрешностей:

$$\gamma_R = \gamma_{K\phi} + \gamma_{\rho s} + \gamma_{Rt} + \gamma_{Rcm} + \gamma_{Rк}, \qquad (8)$$

где $\gamma_{K\phi}$ – погрешность коэффициента формы $\gamma_{К\phi} = \frac{\Delta l}{l} + \frac{\Delta b}{b}$; $\gamma_{\rho s}$ – погрешность воспроизведения величины ρ_s резистивной пленки; γ_{Rt} – температурная погрешность; γ_{Rcm} – погрешность, обусловленная старением пленки; $\gamma_{Rк}$ – погрешность переходных сопротивлений контактов [1].

Погрешность воспроизведения удельного поверхностного сопротивления $\gamma_{\rho s}$ зависит от условий напыления и материала резистивной пленки. В условиях серийного производства её значение не превышает 5 %, $\gamma_{\rho s} = 5\ \%$.

Температурная погрешность зависит от *TKR* материала пленки:

$$\gamma_{Rt} = \alpha_R \cdot (T_{\max} - T_{\min}), \qquad (9)$$

где α_R – температурный коэффициент сопротивления материала пленки, $T_{\max} - T_{\min}$ – рабочий диапазон температур.

Погрешность $\gamma_{Rст}$, обусловленная старением пленки, вызвана медленным изменением структуры пленки во времени и её окислением. Она зависит от материала пленки и эффективности защиты, а также от условий хранения и эксплуатации. Обычно для ГИМС $\gamma_{Rст}$ не превышает 3 %.

Погрешность переходных сопротивлений контактов $\gamma_{Rк}$ зависит от технологических условий напыления пленок, удельного сопротивления резистивной пленки и геометрических размеров контактного перехода: длины перекрытия контактирующих пленок, ширины резистора. Обычно $\gamma_{Rк} = 1 \div 2$ %.

Из (8) определяем $\gamma_{Кф}$, которая должна быть больше нуля. Если $\gamma_{Кф} < 0$, необходимо выбрать другой материал, с меньшими значениями остальных составляющих погрешности.

5. Определяется конструкция резисторов по значению коэффициента формы $k_ф$:

$$k_{фi} = R_i / \rho_s. \qquad (10)$$

Если $1 < k_{фi} < 10$, то рекомендуется конструировать резистор прямоугольной формы; при $0{,}1 < k_{фi} < 1$ – использовать резистор прямоугольной формы, у которого длина меньше ширины; $k_{фi} > 10$ – резистор сложной формы.

6. Рассчитываются прямоугольные полосковые тонкопленочные резисторы.

Выбираем технологию. Определяем ограничения для выбранной технологии. Далее проводим расчет последовательно для каждого резистора.

Для резисторов, имеющих $1 < k_{фi} < 10$, расчетное значение ширины резистора определяется из условия

$$b_{\text{расч}} > \max \{ b_{\text{техн}};\ b_{\text{точн}};\ b_{\text{р}} \}, \qquad (11)$$

где $b_{\text{расч}}$ – минимальная ширина резистора, определяемая возможностями выбранного технологического процесса;
$b_{\text{точн}}$ – ширина резистора, определяемая точностью изготовления:

$$b_{\text{точн}} > (\Delta b + \Delta l / k_{\text{ф}}) / \gamma_{K\phi\ \text{доп}}, \quad (12)$$

где Δb, Δl – погрешности изготовления ширины и длины резистора, зависящие от выбранного метода изготовления;

$b_{\text{р}}$ – минимальная ширина резистора, при которой рассеивается заданная мощность,

$$b_p = \sqrt{\frac{P}{P_0 \cdot k_\phi}}. \quad (13)$$

За ширину b резистора принимается ближайшее к $b_{\text{расч}}$ наибольшее значение, кратное шагу координатной сетки, принятому для чертежа топологии с учетом масштаба. Для тонкопленочной технологии шаг координатной сетки обычно составляет 0,01 или 0,005 мм.

Расчетная длина резистора определяется по формуле

$$l_{расч} = b \cdot k_\phi. \quad (14)$$

Полная длина резистора с учетом перекрытия контактных площадок определяется выражением:

$$l_{полн} = l + 2e, \quad (15)$$

где e – размер перекрытия резистора и контактной площадки (e = 0,1 мм).

Площадь, занимаемая резистором на площадке, определяется по формуле

$$S = l_{полн} \cdot b. \quad (16)$$

Для проверки правильности расчета находятся действительная удельная мощность рассеивания и погрешность резистора. Резистор спроектирован удовлетворительно, если:

1) удельная мощность рассеивания

$$P^*_0 = P/S < P_0; \quad (17)$$

2) погрешность коэффициента формы $\gamma^*_{Кф}$ не превышает допустимого значения $\gamma_{Кф}$:

$$\gamma^*_{\text{Кф}} = \Delta l / l_{полн} + \Delta b / b \leq \gamma_{\text{Кф доп}}; \quad (18)$$

3) суммарная погрешность γ^*_{R} не превышает допуска γ_{R}:

$$\gamma^*_{\text{R}} = \gamma_{\text{Кф}} + \gamma_{\rho\text{s}} + \gamma_{\text{Rt}} + \gamma_{\text{Rст}} + \gamma_{\text{Rк}} \leq \gamma_{\text{R}}, \quad (19)$$

где $\gamma_{\text{Кф}} = \Delta l / l + \Delta b / b$.

Для резисторов с $0{,}1 < k_{фi} < 1$ расчет начинаем с определения длины, а затем определяем ширину.

Расчетное значение длины резисторов $l_{расч}$ выбирается из условия

$$l_{расч} > \max \{ l_{техн}; l_{точн}; l_p \}, \quad (20)$$

где $l_{расч}$ – минимальная длина резистора, определяемая разрешающей способностью выбранного технологического метода формирования конфигурации; $l_{точн}$ – минимальная длина резистора, при которой обеспечивается заданная точность:

$$l_{\text{точн}} > (\Delta l + \Delta b \cdot k_{\text{ф}}) / \gamma_{\text{Кф доп}}; \quad (21)$$

l_p – минимальная длина резистора, при которой рассеивается заданная мощность:

$$l_{pi} = \sqrt{\frac{P \cdot K_{\text{ф}i}}{P_0}}. \quad (22)$$

Для метода фотолитографии $l_{техн} = 0{,}1$ мм.

Расчетная ширина резисторов определяется по формуле

$$b_{\text{расч } i} = \frac{l}{K_{\text{ф}i}}. \quad (23)$$

Полная длина резистора с учетом перекрытия контактных площадок и площадь резистора определяются соответственно по формулам (15, 16).

Далее проведем расчет резисторов с $K_ф > 10$. Такие резисторы будем выполнять в форме меандра. Ширина резистора рассчитывается так же, как для резисторов с $1 < k_{фi} < 10$.

Длина средней линии меандра

$$l_{ср} = b_{расч} \cdot K_ф. \quad (24)$$

Задаемся расстоянием между соседними звеньями меандра a_{min}. Для метода фотолитографии a_{min} берется из технологических ограничений, для масочного метода обычно задают $a_{min} = b_{расч}$. Таким образом, шаг одного звена $t = 2 \cdot b_{расч}$ или $t = a_{min} + b_{расч}$.

Оптимальное число звеньев меандра

$$n_{опт} = \sqrt{\frac{l_{ср}}{t} \cdot \frac{B}{L}}. \quad (25)$$

При меандре квадратной формы $L=B$:

$$n_{опт} = \sqrt{K_{ф2}/_2}. \quad (26)$$

Определим длину меандра:

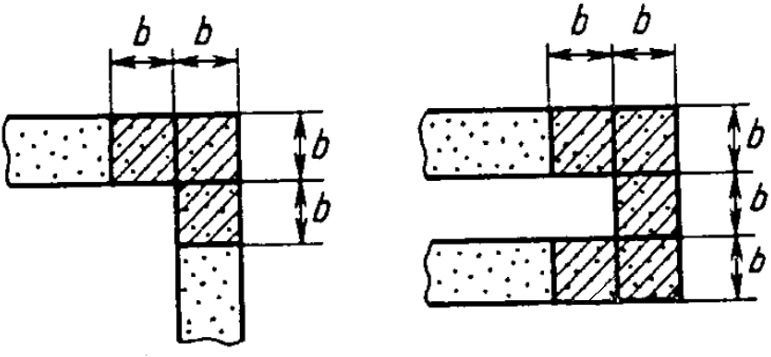
$$L = n_{опт} \cdot (a + b_{расч}). \quad (27)$$

Определим ширину меандра:

$$B = \frac{l_{ср} - a \cdot n_{опт}}{n_{опт}}. \quad (28)$$

Также необходимо учитывать, что при использовании масочного метода должно выполняться соотношение $B_{max} / a \leq 10$, обеспечивающее требуемую жесткость маски.

Приведенные расчетные соотношения не учитывают, что в резисторах типа «меандр» плотность тока в изгибах неравномерна. Неравномерное распределение плотности тока наблюдается в пределах трех квадратов области изгиба (рис. 3).



а б

Рис. 3. Конструкции изгибов резисторов типа «меандр»: а – Г-образный изгиб; б – П-образный изгиб [1]

Сопротивление П-образных изгибов $R_{\text{и}} = 4 \cdot \rho_{s_{\text{опт}}}$; Г-образных - $R_{\text{и}} = 2{,}55 \cdot \rho_{s_{\text{опт}}}$. Для уточненного расчета с учетом изгибов конструкцию резистора представляют в виде последовательно соединенных прямолинейных участков и изгибов. Общее сопротивление будет определяться как

$$R_i = R_{\text{и}} m + \frac{l_{\text{п}} n}{b} \rho_s, \tag{29}$$

где $R_{\text{и}}$ – сопротивление изгибов; m – количество изгибов; $l_{\text{п}}$ – длина прямолинейных участков; n – число звеньев меандра.

Длина прямолинейного участка одного звена меандра

$$l_{\text{п}} = \frac{R_i - m \cdot R_{\text{и}}}{\rho_s \cdot n} \cdot b. \tag{30}$$

После этого корректируют размеры L и B с целью обеспечения заданного номинала резистора и проводят проверку.

7. Рассчитывается суммарная площадь всех резисторов.

2. Последовательность выполнения работы

1. Изучите теоретическую часть работы.

2. Получите задание в виде принципиальной схемы у преподавателя.

3. Согласно принципиальной схеме проведите расчеты мощностей, выделяемых на резисторах.

4. Выберите материал резистивной пленки и контактных площадок.

5. Рассчитайте конструкции всех резисторов.

6. Сделайте проверку расчетов.

7. Определите суммарную площадь, занимаемую всеми резисторами.

3. Содержание отчета

1. Цель работы.

2. Результаты расчетов.

3. Чертежи тонкопленочных резисторов в масштабе с указанием размеров.

4. Выводы по полученным результатам.

Контрольные вопросы

1. Что такое гибридная интегральная схема?

2. Назовите требования, предъявляемые к материалам резистивных пленок.

3. Назовите основные материалы резистивных пленок ГИМС.

4. Назовите требования, предъявляемые к материалам контактных площадок.

5. Назовите основные материалы контактных площадок ГИМС.

6. Какие конструкции бывают у тонкопленочных резисторов?

7. Что такое коэффициент формы тонкопленочного резистора?

8. Порядок расчета резистора с коэффициентом формы от 1 до 10.

9. Порядок расчета резистора с коэффициентом формы от 0,1 до 1.

10. Порядок расчета резистора с коэффициентом формы больше 10.

11. Расскажите о подстраиваемых резисторах.

12. Способы подгонки номинала резистора.

Библиографический список

1. Коледов Л.А. Конструирование и технология микросхем: учеб. пособие для вузов. М.: Высшая школа, 1984. 232 с.

2. Меркулов А.И., Меркулов В.А. Основы конструирования интегральных микросхем. Самара: Издательство СГАУ, 2013. 270 с.

3940

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПРИМЕНЕНИЕ ЯЗЫКА ОПИСАНИЯ АППАРАТУРЫ VHDL ДЛЯ ПРОЕКТИРОВАНИЯ УСТРОЙСТВ ЦИФРОВОЙ СХЕМОТЕХНИКИ НА ПРИМЕРЕ ПОСТРОЕНИЯ ПРИЕМОПЕРЕДАТЧИКОВ ПО СТАНДАРТУ RS232

Методические указания к лабораторным работам

Рязань 2007

УДК 621.315.6

Применение языка описания аппаратуры VHDL для проектирования устройств цифровой схемотехники на примере построения приемо-передатчиков по стандарту RS232: методические указания к лабораторным работам / Рязан. гос. радиотехн. ун-т; сост. С.В. Челебаев. – Рязань: РГРТУ, 2007. – 16 с.

Изложены краткие теоретические сведения о последовательном интерфейсе RS232 и его аппаратных средствах. Приведен маршрут проектирования цифровых устройств на базе ПЛИС. Описан подход к программированию ПЛИС, базирующийся на применении языка описания аппаратуры VHDL. Рассмотрен способ проектирования передатчика и приемника по стандарту RS232 на основе конечного автомата с его последующей реализацией на VHDL.

Предназначены для студентов специальности 200400.

Табл. 1 . Ил.16 . Библиогр.: 2 назв.

Проектирование, цифровые схемы, RS232, VHDL, Xilinx ISE, САПР

Печатается по решению редакционно-издательского совета Рязанского государственного радиотехнического университета.

Рецензент: кафедра БМПЭ РГРТУ (зав. кафедрой С.В. Вихров)

Применение языка описания аппаратуры VHDL
для проектирования устройств цифровой схемотехники
на примере построения приемопередатчиков по стандарту RS232

Составитель Ч е л е б а е в Сергей Валерьевич

Редактор Н.А. Орлова
Корректор С.В. Макушина

Подписано в печать 20.04.07. Формат бумаги 60x84 1/16.
Бумага газетная. Печать трафаретная. Усл. печ. л. 1,0.
Уч.-изд. л. 1,0. Тираж 40 экз. Заказ

Рязанский государственный радиотехнический университет.
390005, Рязань, ул. Гагарина, 59/1.
Редакционно-издательский центр РГРТУ.

ЛАБОРАТОРНАЯ РАБОТА № 1

Разработка передатчика байтов по стандарту RS232

Краткие теоретические сведения

Последовательный интерфейс RS232

RS232 – последовательный интерфейс, в котором данные передаются побитно по очереди один за другим по единственному проводнику. Каждый байт передается вместе со служебными битами (рис. 1). К служебным битам относятся стартовый бит, стоповый бит и биты четности. Как правило, стартовый бит кодируется логическим нулем, а стоповый – логической единицей. Стартовый и стоповый биты являются обязательными. Биты четности являются необязательными. Их применение определяется протоколом взаимодействия между передающим и приемным устройствами. Число передаваемых бит в байте может быть меньше восьми.

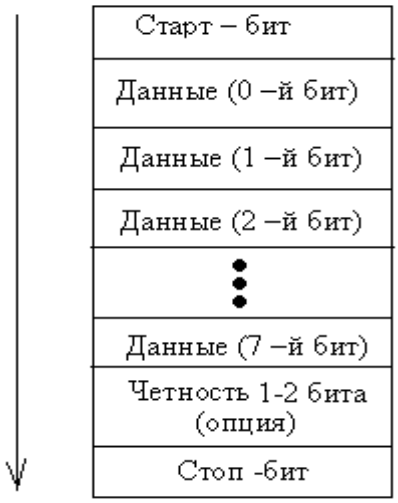


Рис. 1. Формат передачи по стандарту RS232

Добавление бита четности является простым методом обнаружения ошибок при передаче данных. Один дополнительный разряд прибавляется для того, чтобы можно было сделать четным или нечетным полное количество единиц или нулей в передаваемой последовательности битов. Одна ошибка изменит число единиц или нулей с четного числа на нечетное или наоборот, и таким образом ошибка будет обнаружена.

Существует несколько режимов четности:

1) четность;
   parity=D0 xor D1 xor D2… xor D7,
   Di – бит данных;
2) нечетность;
   parity=D0 xnor D1 xnor D2… xnor D7,
   где xnor(D0, D1) = not(xor(D0, D1));
3) программное задание четности.

При программном задании четности вместо бита четности передается ноль либо единица.

Для обнаружения двойной ошибки используются двухбитные режимы четности.

Во время отсутствия передачи в линию передается стоп-бит.

На рис. 2 показаны временные диаграммы передачи одного байта с одним битом четности (Т – время передачи одного бита).

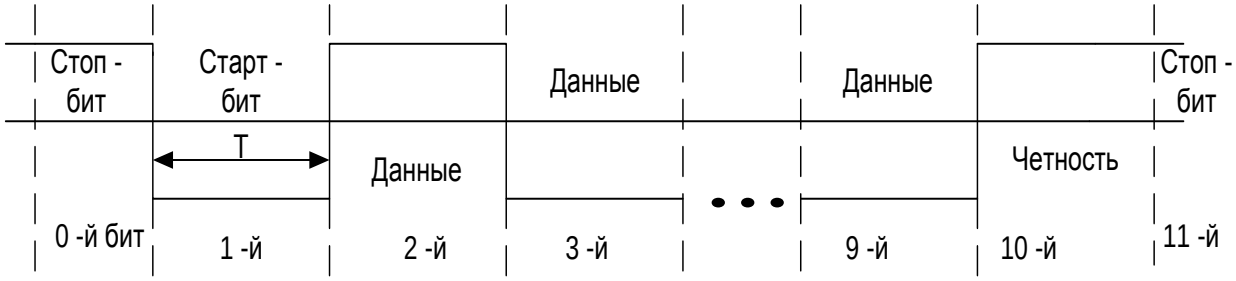


Рис. 2. Временные диаграммы передачи байта с одним битом четности

Число изменений состояния канала передачи данных в секунду называется скоростью передачи в бодах. В системе с двумя возможными состояниями скорость передачи в бодах равна числу битов, передаваемых за секунду (бит/с).

При пересылке данных между двумя устройствами по стандарту RS232 оба устройства должны работать в одном и том же режиме так, чтобы во время передачи какого-либо байта устройство-приемник принимало бы данные на той же скорости, на которой их посылал передатчик. Такая система является асинхронной, так как минимальное число символов, передаваемых в секунду, определяется скоростью работы передатчика, т.е. передатчик может сделать паузу любой длительности между символами. Во время любых пауз линия находится в состоянии логической единицы, поэтому естественно в качестве стартового бита при передаче использовать логический ноль.

На рис. 3 показаны временные диаграммы передачи байта с кодом 0x55.

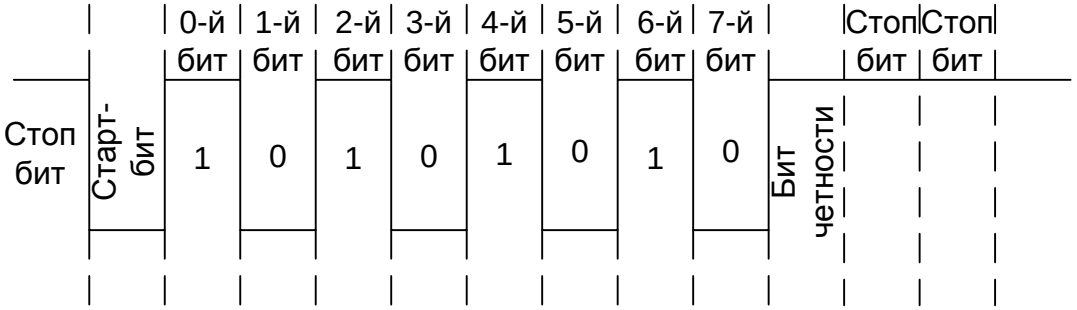


Рис. 3. Временные диаграммы передачи кода 0x55 посредством RS232

Аппаратные средства последовательного интерфейса

Специальные ИС, называемые универсальными асинхронными приемо-передатчиками (universal asynchronous receiver/transmitters – UART), преобразуют данные из параллельного формата, получаемого от шины данных микропроцессора или спецвычислителя, в последовательный. Эти устройства можно рассматривать как устройства с двумя внутренними секциями: преобразователем параллельного формата в последовательный, называемым приемо-передатчиком, и преобразователем последовательного формата в параллельный, называемым приемником. Передатчик подставляет бит четности, стартовый и стоповые биты. Приемник проверяет формат и четность полученных данных. Приемник и передатчик могут работать одновременно, такой способ передачи называется полным дуплексом. Передача только в одном направлении называется симплексной связью. Передача

в обоих направлениях, но не в одно и то же время, называется полудуплексной связью.

Скорости передачи и приема данных контролируются внутренним или внешним генератором тактовых импульсов, работающим на частоте, кратной скорости передачи данных.

Приемопередатчик по стандарту RS232 может быть реализован с применением программируемых логических интегральных схем (ПЛИС). При этом ПЛИС осуществляет преобразование данных из параллельного формата в последовательный и наоборот, а преобразование уровней логических сигналов, подаваемых в линию передачи данных, осуществляет микросхема – преобразователь уровня, например, MAX212.

Маршрут проектирования цифровых устройств на базе ПЛИС фирмы Xilinx

Для создания цифрового устройства на базе ПЛИС Xilinx необходимо выполнить следующую последовательность операций:

1) создать новый проект, указав серию, тип ПЛИС и средств синтеза;
2) разработать описание проектируемого устройства в схемотехнической (с использованием схемотехнического редактора), алгоритмической (с использованием графа состояний конечного автомата) или текстовой (с использованием языков описания аппаратуры – hardware description languages, HDL) формах;
3) выполнить синтез устройства;
4) провести проверку проекта методом функционального моделирования;
5) выполнить размещение и трассировку проекта в кристалл;
6) провести окончательную верификацию проекта методом временного моделирования;
7) загрузить конфигурационные данные проекта в кристалл (выполнить программирование ПЛИС).

Операции функционального и временного моделирования не являются обязательными, но позволяют значительно сократить общее время разработки устройства за счет раннего обнаружения возможных ошибок.

Программирование ПЛИС с помощью языка VHDL

К современным HDL-языкам относятся VHDL (Very High Speed HDL) и Verilog.

Проект на VHDL состоит из макросов, на основе которых строится описание схемы в целом. Структура макроса имеет вид, показанный на рис. 4.

Цель работы: изучение последовательного интерфейса RS232, построение передатчика байтов по стандарту RS232 как конечного автомата с его последующей реализацией на языке VHDL.

Исходные данные заданы в табл. 1.

Задание:

1) построить автомат передатчика байтов по стандарту RS232;
2) реализовать автомат передатчика байтов на языке VHDL;
3) провести моделирование передатчика с помощью HDL-Bencher;

4) получить конфигурационную последовательность передатчика.

Пп. 1 – 2 задания необходимо выполнить при подготовке к работе.

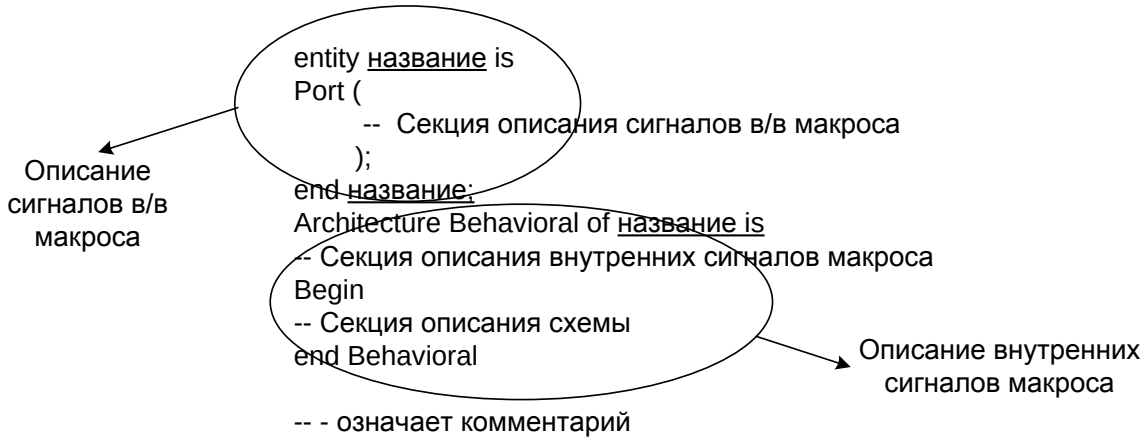


Рис. 4. Структура макроса, написанного на языке VHDL

Таблица 1

Варианты заданий

№ варианта	Тактовая частота F_T	Скорость *Speed* , бит/с	Режим четности
1	25 МГц	2 500 000	четность
2	12,5 МГц	2 500 000	нечетность
3	60 МГц	10 000 000	четность
4	70 МГц	10 000 000	нечетность
5	1843,2 кГц	230 400	четность
6	2073,6 кГц	230 400	нечетность
7	1267,2 кГц	115 200	четность
8	1382,4 кГц	115 200	нечетность
9	748,8 кГц	57 600	четность
10	806,4 кГц	57 600	нечетность
11	432 кГц	28 800	четность
12	460,8 кГц	28 800	нечетность
13	224,8 кГц	14 400	четность
14	259,2 кГц	14 400	нечетность
15	729,6 кГц	38 400	четность
16	768 кГц	38 400	нечетность
17	403,2 кГц	19 200	четность
18	422,4 кГц	19 200	нечетность
19	220,8 кГц	9 600	четность
20	230,4 кГц	9 600	нечетность

Содержание отчета:

1) название, цель работы, задание;
2) схема автомата передатчика по стандарту RS232;
3) программа передатчика на языке VHDL;
4) результаты моделирования схемы.

Пример выполнения работы

Исходные данные: тактовая частота – 4 МГц, скорость – 1 000 000 бит/с, режим четности – отсутствует.

1-й этап. Построение автомата передатчика байтов по стандарту RS232 как конечного автомата Мили.

Передатчик байта как конечный автомат должен иметь следующие состояния:
1) начальное состояние;
2) состояние передачи стартового бита;
3-10) состояние передачи i -го бита данных ($i = \overline{0,\ldots,7}$);
11) состояние передачи бита четности (при наличии бита четности);
12) состояние передачи стопового бита.

При построении автомата необходимо определить число тактов, требуемых для передачи одного бита данных:

$$kol_takt = F_{\text{Т}} / Speed .$$

Для рассматриваемого примера $kol_takt = 4$.

Автомат передатчика примет вид, показанный на рис. 5.

На графе автомата (рис. 5) введены следующие обозначения:
DATA(7 DOWNTO 0) – передаваемый байт;
cnt_takt – счетчик тактов, затрачиваемых на передачу одного бита;
TX – передаваемый в линию передачи данных бит;
Reset – сигнал сброса автомата в начальное состояние;
Start – команда начала передачи байта.

2-й этап. Реализация автомата передатчика байтов по стандарту RS232 на языке VHDL.

Для запуска пакета *Xilinx ISE* в меню программы следует выбрать группу Xilinx ISE, в которой нужно выделить строку *Project Navigator* и щелкнуть на ней левой кнопкой мыши. При этом на экране монитора появится основное окно Навигатора проекта (рис. 6), которое содержит кроме стандартных элементов 4 встроенных окна:
– окно исходных модулей проекта (*Sources in Project*);
– окно необходимых процессов для выбранного исходного модуля (*Processes for Current Source*);
– окно консольных сообщений программных модулей (*Console*);
– окно редактора текстовых HDL-описаний проекта.

В окне исходных модулей проекта (рис. 6) отображается иерархическая структура, состоящая из модулей, в которых содержится описание проектируемого устройства и описание тестовых воздействий, используемых в процессе моделирования. Каждый тип модуля имеет соответствующее графическое обозначение – пиктограмму.

Окно процессов (рис. 6) показывает маршрут обработки выделенного исходного модуля в процессе проектирования устройства. В данном окне подробно отображаются все этапы процесса разработки и программирования ПЛИС. Последовательность и содержание этапов определяются типом исходного модуля и серией ПЛИС.

Окно консольных сообщений (рис. 6) предназначено для вывода информации программных модулей пакета, работающих в консольном режиме. Здесь отображаются сообщения об ошибках и предупреждения.

Окно интегрированного текстового редактора (рис. 6) становится активным, если для проектируемого устройства или используемых библиотек выбран способ описания на языке HDL.

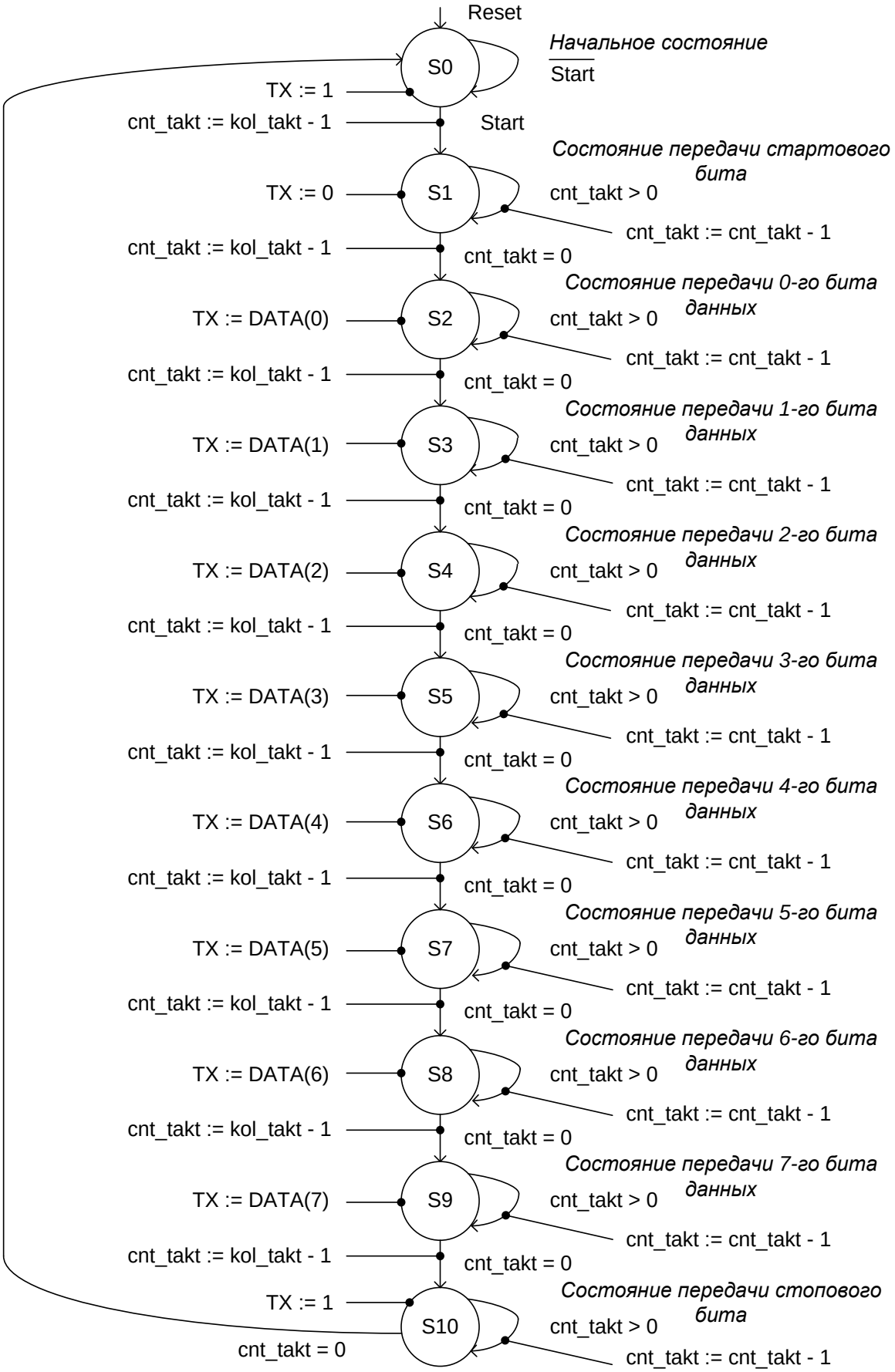


Рис. 5. Передатчик байтов по стандарту RS232 как автомат Мили

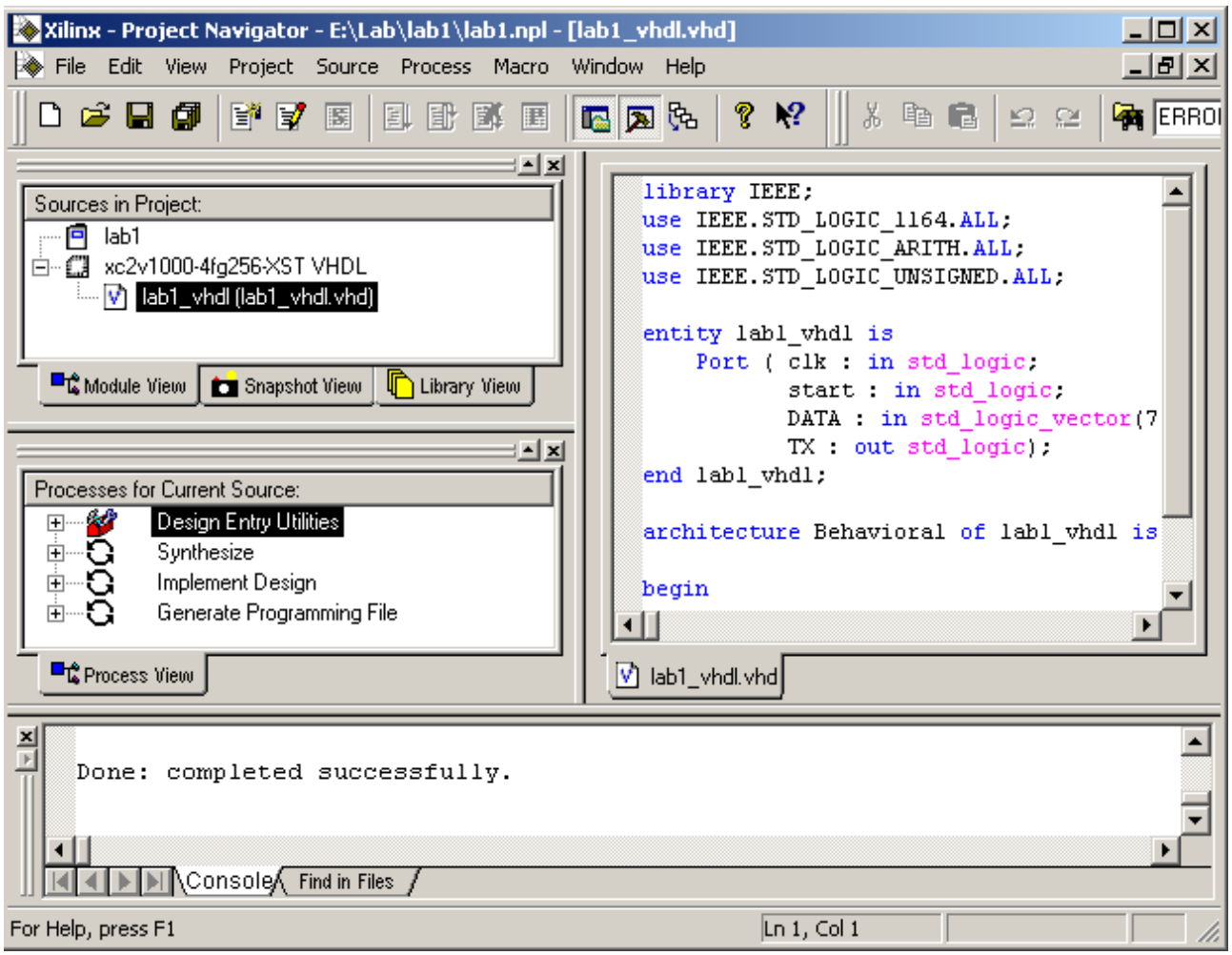


Рис. 6. Основное окно *Навигатора проекта* пакета *Xilinx ISE*

Для создания нового проекта следует выполнить команду *File* основного меню *Навигатора проекта*, а затем во всплывающем меню выбрать строку *New Project*. При этом появится диалоговое окно, показанное на рис. 7.

В открывшейся диалоговой панели (рис. 7) нужно определить исходные данные, необходимые для создания проекта:

– название проекта (*Project Name*);
– диск и каталог, в котором предполагается расположить проект (*Project Location*);
– серию ПЛИС, на базе которой разрабатывается устройство (*Device Family*);
– тип кристалла (*Device*);
– корпус;
– быстродействие;
– средство синтеза (*Design Flow*).

Для создания проекта на VHDL необходимо в меню *Design Flow* (рис. 7) выбрать строку *XST VHDL*.

Процедура определения всех необходимых параметров создаваемого проекта завершается нажатием клавиши *OK*, находящейся в нижней части панели установки параметров нового проекта (рис. 7).

Для создания нового модуля исходного описания проекта следует выбрать команду *New Source* из раздела *Project* основного меню. В открывшейся диалоговой панели, показанной на рис. 8, необходимо выбрать тип нового модуля, записать его имя и указать место расположения файла на диске.

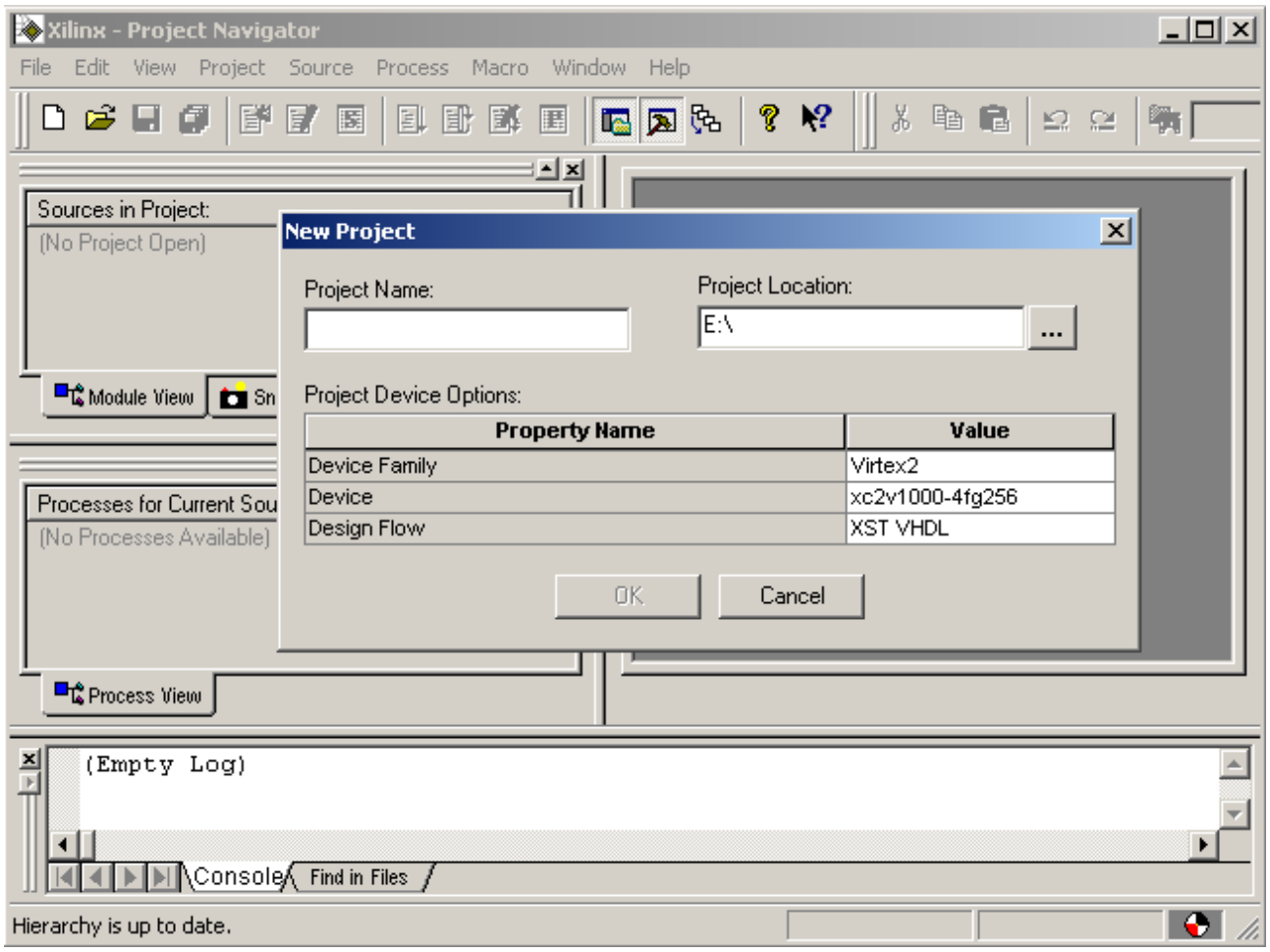


Рис. 7. Создание нового проекта в САПР Xilinx ISE

Для разработки функциональной схемы проекта следует в предложенном списке диалоговой панели выбрать тип создаваемого исходного модуля *VHDL Module*, щелкнув на соответствующей строке левой кнопкой мыши. Затем нужно активизировать поле редактирования названия модуля (файла) *File Name* и ввести текст имени с помощью клавиатуры. Расширение имени файла устанавливается автоматически в соответствии с выбранным типом модуля. Место расположения создаваемого модуля на диске указывается в поле редактирования *Location* диалоговой панели (рис. 8). По умолчанию предлагается рабочий каталог текущего проекта. Если флаг индикатора *Add to project* находится в установленном состоянии, то создаваемый модуль автоматически включается в состав текущего проекта. Установка значений всех необходимых параметров создаваемого модуля завершается нажатием клавиши *Next* (*Далее*), которая находится в нижней части диалоговой панели (рис. 8). При этом открывается информационная панель *Define VHDL Source*, показанная на рис. 9.

В диалоговом окне *Define VHDL Source* (рис. 9) задаются название модуля (поле *Entity name*), имя архитектуры (*Architecture Name*), а также входные и выходные сигналы модуля. В поле *Port Name* указываются названия входов и выходов, в поле *Direction* – их направление (*in* или *out*). Поля *MSB* и *LSB* задают разрядность многобитных сигналов (шин). На рис. 9 показано задание входных и выходных сигналов для передатчика байтов по стандарту RS232. Следует отметить, что сигналы ввода-вывода в диалогом окне (рис. 9) вводить не обязательно. Их можно добавить на этапе разработки HDL-описания. По завершении установок

параметров в окне *Define VHDL Source* (рис. 9) следует нажать клавишу *Далее*. В результате этого появится окно *New Source Information* (рис. 10).

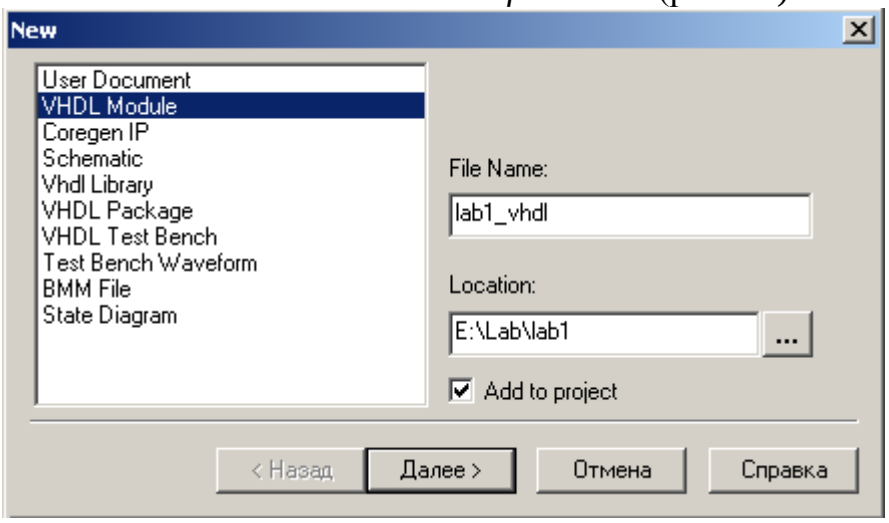


Рис. 8. Диалоговая панель установки параметров нового исходного модуля проекта

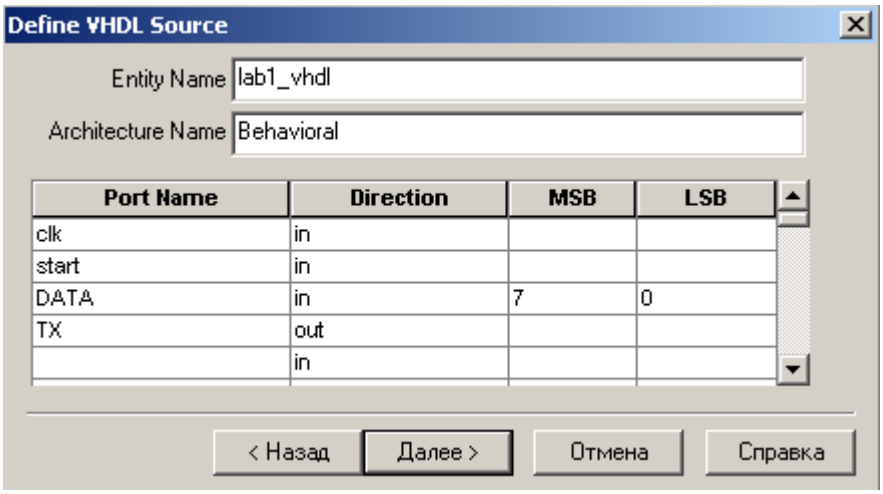


Рис. 9. Диалоговое окно задания сигналов ввода-вывода исходного модуля

Для перехода к редактированию HDL-описания в окне (рис. 10) необходимо нажать клавишу *Готово*.

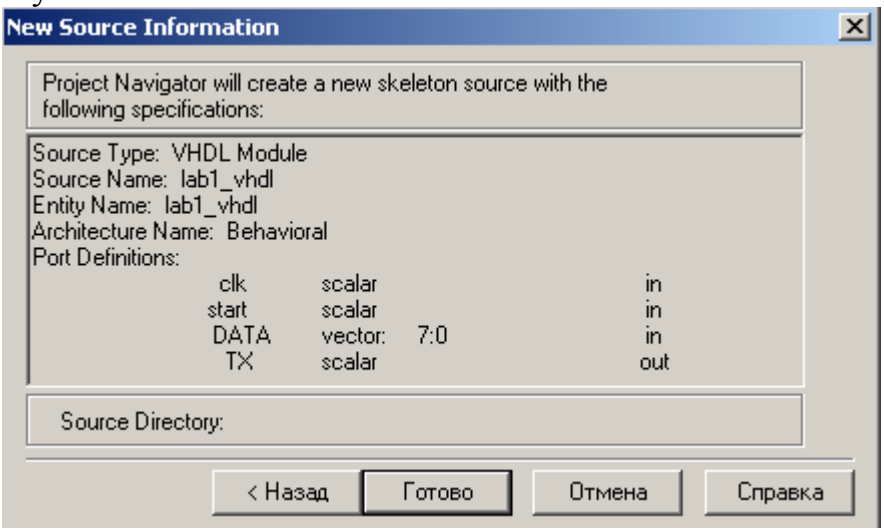


Рис. 10. Окно информации о новом проекте

Разработанная программа на языке VHDL приняла вид:

```
library IEEE; -- Раздел описания библиотек
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity lab1_vhdl is  -- Раздел объявления входных и выходных сигналов
   Port ( clk   : in std_logic;  -- тактовая частота
          start : in std_logic;  -- сигнал запуска передачи байта
          DATA  : in std_logic_vector(7 downto 0);  -- код передаваемого байта
          TX    : out std_logic;  -- выходной сигнал, посылаемый в линию передачи байта
          S0_O  : out std_logic;  -- выходные состояния автомата, выведенные на выходные
          S1_O  : out std_logic;  -- ножки с целью их просмотра в симуляторе
          S2_O  : out std_logic;
          S3_O  : out std_logic;
          S4_O  : out std_logic;
          S5_O  : out std_logic;
          S6_O  : out std_logic;
          S7_O  : out std_logic;
          S8_O  : out std_logic;
          S9_O  : out std_logic;
          S10_O : out std_logic
         );
end lab1_vhdl;

architecture Behavioral of lab1_vhdl is  -- описание передатчика байтов

  SIGNAL S0                : STD_LOGIC;  -- состояние автомата S0
  SIGNAL S1                : STD_LOGIC;  -- состояние автомата S1
  SIGNAL S2                : STD_LOGIC;  -- состояние автомата S2
  SIGNAL S3                : STD_LOGIC;  -- состояние автомата S3
  SIGNAL S4                : STD_LOGIC;  -- состояние автомата S4
  SIGNAL S5                : STD_LOGIC;  -- состояние автомата S5
  SIGNAL S6                : STD_LOGIC;  -- состояние автомата S6
  SIGNAL S7                : STD_LOGIC;  -- состояние автомата S7
  SIGNAL S8                : STD_LOGIC;  -- состояние автомата S8
  SIGNAL S9                : STD_LOGIC;  -- состояние автомата S9
  SIGNAL S10               : STD_LOGIC;  -- состояние автомата S10

  SIGNAL NEW_S0            : STD_LOGIC;  -- значение состояния S0 в следующем такте
  SIGNAL NEW_S1            : STD_LOGIC;  -- значение состояния S1 в следующем такте
  SIGNAL NEW_S2            : STD_LOGIC;  -- значение состояния S2 в следующем такте
  SIGNAL NEW_S3            : STD_LOGIC;  -- значение состояния S3 в следующем такте
  SIGNAL NEW_S4            : STD_LOGIC;  -- значение состояния S4 в следующем такте
  SIGNAL NEW_S5            : STD_LOGIC;  -- значение состояния S5 в следующем такте
  SIGNAL NEW_S6            : STD_LOGIC;  -- значение состояния S6 в следующем такте
  SIGNAL NEW_S7            : STD_LOGIC;  -- значение состояния S7 в следующем такте
  SIGNAL NEW_S8            : STD_LOGIC;  -- значение состояния S8 в следующем такте
  SIGNAL NEW_S9            : STD_LOGIC;  -- значение состояния S9 в следующем такте
  SIGNAL NEW_S10           : STD_LOGIC;  -- значение состояния S10 в следующем такте

  SIGNAL Reset             : STD_LOGIC;  -- сигнал сброса автомата
  SIGNAL cnt_takt          : STD_LOGIC_VECTOR(2 DOWNTO 0);  -- счетчик тактов
  SIGNAL kol_takt          : STD_LOGIC_VECTOR(2 DOWNTO 0);  -- количество тактов,
    -- затрачиваемых на передачу одного бита данных

  SIGNAL CNT_TAKT_EQ_0  : STD_LOGIC;  -- сигнал равенства нулю счетчика тактов
```

```
  SIGNAL LOGIC_1              : STD_LOGIC;  -- сигнал, используемый для сброса
  SIGNAL LOGIC_1_T            : STD_LOGIC;  -- сигнал, используемый для сброса
begin  -- начало программы
  kol_takt <= "100";  -- значение задается в двоичной сист. числен. в соответствии с заданием
-- формирование значений состояний автомата в следующем такте
  NEW_S0 <= Reset OR (S0 AND (NOT START)) OR (S10 AND CNT_TAKT_EQ_0);
  NEW_S1 <= (S0 AND START) OR (S1 AND (NOT CNT_TAKT_EQ_0));
  NEW_S2 <= (S1 AND CNT_TAKT_EQ_0) OR (S2 AND (NOT CNT_TAKT_EQ_0));
  NEW_S3 <= (S2 AND CNT_TAKT_EQ_0) OR (S3 AND (NOT CNT_TAKT_EQ_0));
  NEW_S4 <= (S3 AND CNT_TAKT_EQ_0) OR (S4 AND (NOT CNT_TAKT_EQ_0));
  NEW_S5 <= (S4 AND CNT_TAKT_EQ_0) OR (S5 AND (NOT CNT_TAKT_EQ_0));
  NEW_S6 <= (S5 AND CNT_TAKT_EQ_0) OR (S6 AND (NOT CNT_TAKT_EQ_0));
  NEW_S7 <= (S6 AND CNT_TAKT_EQ_0) OR (S7 AND (NOT CNT_TAKT_EQ_0));
  NEW_S8 <= (S7 AND CNT_TAKT_EQ_0) OR (S8 AND (NOT CNT_TAKT_EQ_0));
  NEW_S9 <= (S8 AND CNT_TAKT_EQ_0) OR (S9 AND (NOT CNT_TAKT_EQ_0));
  NEW_S10 <= (S9 AND CNT_TAKT_EQ_0) OR (S10 AND (NOT CNT_TAKT_EQ_0));
-- описание триггеров, хранящих значения состояний (для способа кодирования ONE HOT)
PROCESS(clk)
BEGIN
 IF (clk='1' and clk'event) THEN  -- условие синхронной операции по переднему фронту
  IF (Reset = '1')  -- сброс автомата
   THEN S0  <= '1';
            S1  <= '0';
            S2  <= '0';
            S3  <= '0';
            S4  <= '0';
            S5  <= '0';
            S6  <= '0';
            S7  <= '0';
            S8  <= '0';
            S9  <= '0';
            S10 <= '0';
   ELSE S0  <= NEW_S0;
           S1  <= NEW_S1;
           S2  <= NEW_S2;
           S3  <= NEW_S3;
           S4  <= NEW_S4;
           S5  <= NEW_S5;
           S6  <= NEW_S6;
           S7  <= NEW_S7;
           S8  <= NEW_S8;
           S9  <= NEW_S9;
           S10 <= NEW_S10;
  END IF;
 END IF;
END PROCESS;
PROCESS(clk)  -- описание счетчика тактов на передачу бита
BEGIN
 IF (clk='1' and clk'event) THEN
  IF (((S0 = '1') AND (NEW_S1 = '1')) OR ((S1 = '1') AND (NEW_S2 = '1')) OR
      ((S2 = '1') AND (NEW_S3 = '1')) OR ((S3 = '1') AND (NEW_S4 = '1')) OR
```

```
      ((S4 = '1') AND (NEW_S5 = '1')) OR ((S5 = '1') AND (NEW_S6 = '1')) OR
      ((S6 = '1') AND (NEW_S7 = '1')) OR ((S7 = '1') AND (NEW_S8 = '1')) OR
      ((S8 = '1') AND (NEW_S9 = '1')) OR ((S9 = '1') AND (NEW_S10 = '1')))
  THEN cnt_takt <= kol_takt - 1;
  ELSE cnt_takt <= cnt_takt - 1;
 END IF;
END IF;
END PROCESS;

PROCESS(cnt_takt)  -- описание компаратора счетчика тактов с нулем
BEGIN
 IF (cnt_takt = "000")
  THEN CNT_TAKT_EQ_0 <= '1';
  ELSE CNT_TAKT_EQ_0 <= '0';
 END IF;
END PROCESS;

PROCESS(clk)  -- формирование выходного сигнала TX
BEGIN
 IF (clk='1' and clk'event) THEN
  IF ((S0 = '1') OR (S10 = '1'))
   THEN TX <= '1';
   ELSE IF (S1 = '1')
          THEN TX <= '0';
          ELSE IF (S2 = '1')
                 THEN TX <= DATA(0);
                 ELSE IF (S3 = '1')
                        THEN TX <= DATA(1);
                        ELSE IF (S4 = '1')
                               THEN TX <= DATA(2);
                               ELSE IF (S5 = '1')
                                      THEN TX <= DATA(3);
                                      ELSE IF (S6 = '1')
                                             THEN TX <= DATA(4);
                                             ELSE IF (S7 = '1')
                                                    THEN TX <= DATA(5);
                                                    ELSE IF (S8 = '1')
                                                           THEN TX <= DATA(6);
                                                           ELSE TX <= DATA(7);
                                                           END IF;
                                                    END IF;
                                             END IF;
                                      END IF;
                               END IF;
                        END IF;
                 END IF;
          END IF;
  END IF;
 END IF;
END PROCESS;

 LOGIC_1 <= '1';  -- формирования сигнала для сброса автомата при включении питания

PROCESS(clk)  -- формирования сигнала для сброса автомата при включении питания
BEGIN
```

```
 IF (clk='1' and clk'event) THEN
  LOGIC_1_T <= LOGIC_1;
 END IF;
END PROCESS;

PROCESS(clk)  формирования сигнала сброса автомата при включении питания
BEGIN
 IF (clk='1' and clk'event) THEN
  Reset <= LOGIC_1 AND (NOT LOGIC_1_T);
 END IF;
END PROCESS;

 S0_O  <= S0;  -- вывод состояний автомата на ножки схемы
 S1_O  <= S1;
 S2_O  <= S2;
 S3_O  <= S3;
 S4_O  <= S4;
 S5_O  <= S5;
 S6_O  <= S6;
 S7_O  <= S7;
 S8_O  <= S8;
 S9_O  <= S9;
 S10_O <= S10;

end Behavioral;  -- конец программы
```

Для запуска компиляции программы необходимо в окне *Sources in Project* выбрать файл *.vhd, а в окне *Processes for Current Source* двойным щелчком мыши запустить *Synthesize*. При отсутствии ошибок следует перейти к моделированию проекта.

3-й этап. Моделирование передатчика байтов с помощью программы HDL-Bencher.

Для моделирования проекта необходимо создать тестовый вектор. Для этого следует выбрать команду *New Source* из раздела *Project* основного меню. В открывшейся диалоговой панели, показанной на рис. 11, необходимо выбрать тип нового модуля *Test Bench Waveform*, записать его имя (обычно его называют по имени vhdl-файла с добавлением суффикса *_tbw* на конце) и указать место расположения файла на диске.

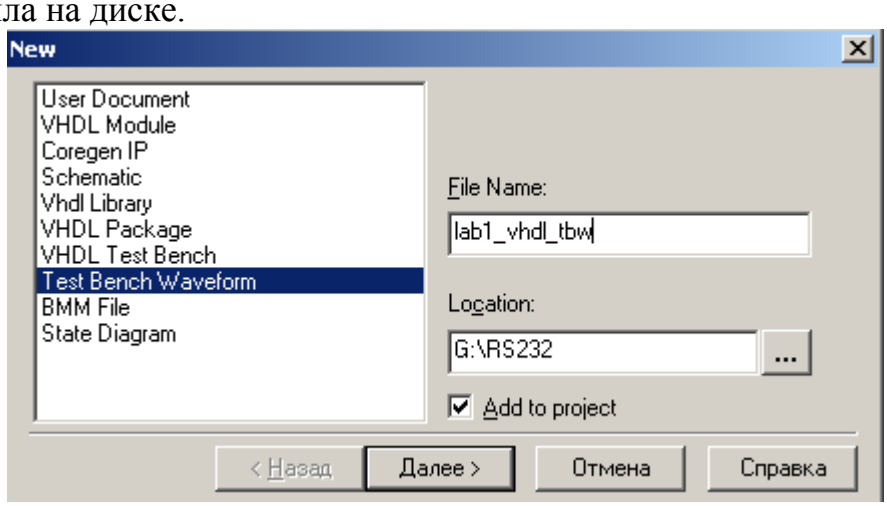


Рис. 11. Диалоговая панель установки параметров тестового вектора исходного модуля проекта

После нажатия клавиши *Далее* появится меню (рис. 12), позволяющее установить взаимосвязь между тестовым вектором и vhdl-модулем.

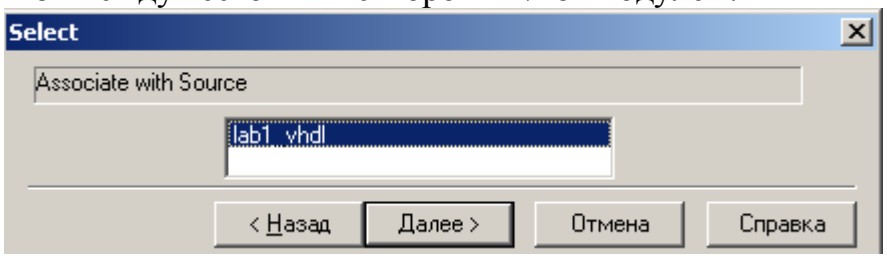


Рис. 12. Диалоговое окно установки взаимосвязи между тестовым вектором и исходным vhdl-описанием

Нажав клавишу *Далее* (см. рис. 12), пользователь увидит окно информации о тестовом векторе (рис. 13).

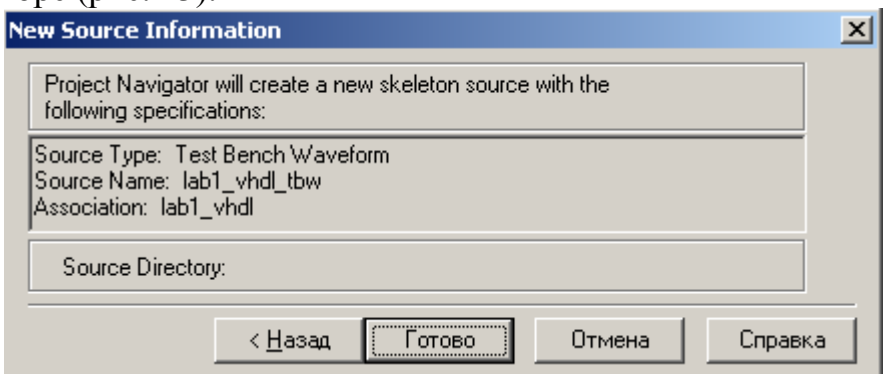


Рис. 13. Окно информации о новом тестовом векторе

После нажатия клавиши *Готово* (см. рис. 13)появится меню временной инициализации *Initialize Timing* (рис. 14). Радиокнопка *Master clock* задает сигнал тактирования. Кнопки *Rising Edge* и *Falling Edge* выбирают способ тактирования элементов памяти по переднему или заднему фронту. Поля *Clock high time* и *Clock low time* определяют длительность единичного и нулевого уровней тактовых сигналов. Для рассматриваемого примера, когда $F_т$=4 МГц, период следования тактовых сигналов составит 250 нс, а длительность их единичного и нулевого уровней – по 125 нс. Поле *Input setup time* задает входную задержку срабатывания запоминающих элементов, а *Output valid delay* – выходную. Поле *Time Scale* задает единицу изменения времени.

После установки всех необходимых параметров меню (рис. 14) следует нажать кнопку *OK*. В результате чего с целью определения входных сигналов запустится *HDL Bencher* (рис. 15). С помощью мыши и клавиатуры необходимо задать входные сигналы. Так, в 3-м такте работы схемы сигнал *start* задан единичным значением, а шина *DATA* – значением 55h. Для изменения масштаба отображения временных диаграмм служат клавиши *Zoom In* и *Zoom Out*. Кнопки *16, 10* и *2* определяют основание системы счисления, в которой отображаются шины. После окончания задания входных сигналов следует нажать клавишу *Safe Waveform*. При этом появляется меню (рис. 16), в поле *End the test bench* которого указывается количество циклов после последнего цикла, в котором осуществлено изменение одного из входных сигналов. После ввода числа нажимается клавиша *OK*. При нажатии *F4* осуществляется выход из программы *HDL Bencher*.

Для запуска проекта на моделирование необходимо в окне *Sources in Project* выбрать файл *.tbw, а в окне *Processes for Current Source* двойным щелчком мыши

во вкладке *ModelSim Simulator* запустить *Generate Expected Simulation Results*. После чего на экране монитора появится окно программы *HDL Bencher* с результатами моделирования.

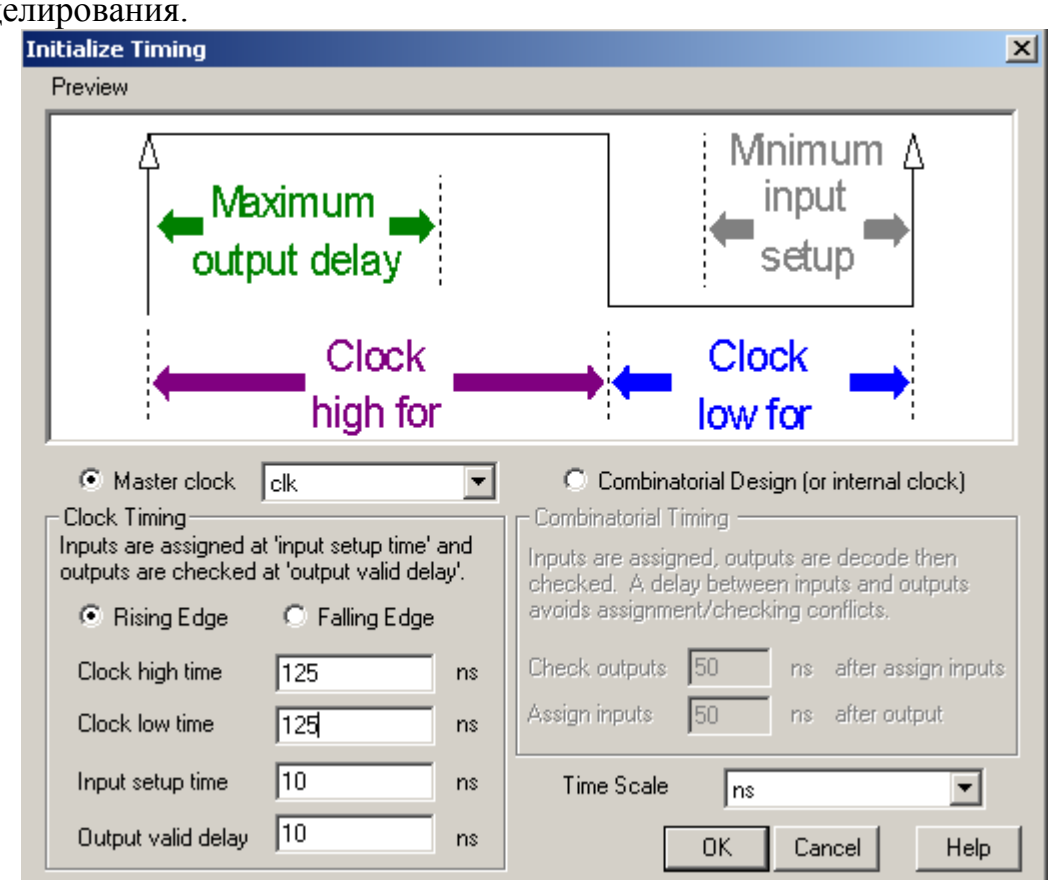


Рис. 14. Меню временной инициализации

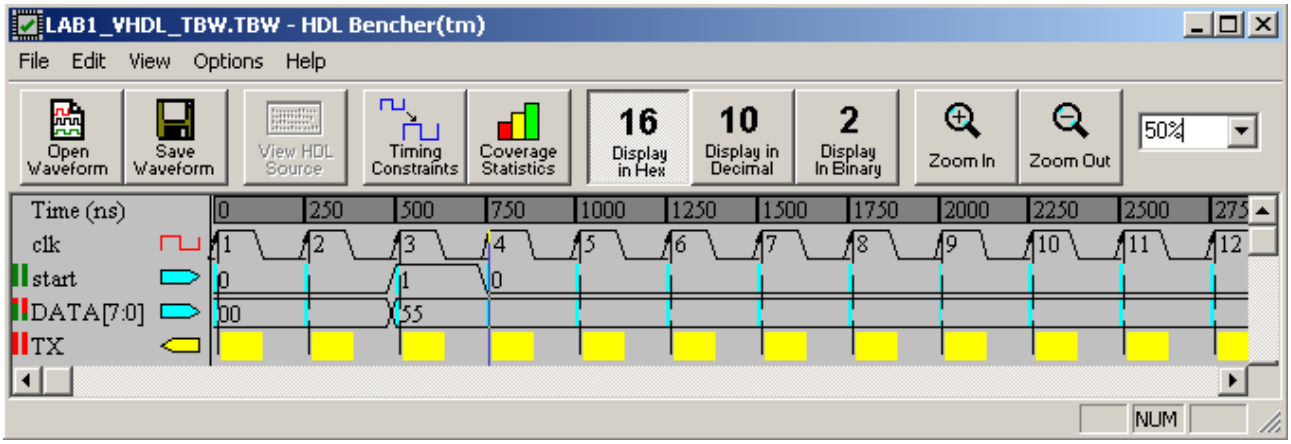


Рис. 15 Окно программы *HDL Bencher*

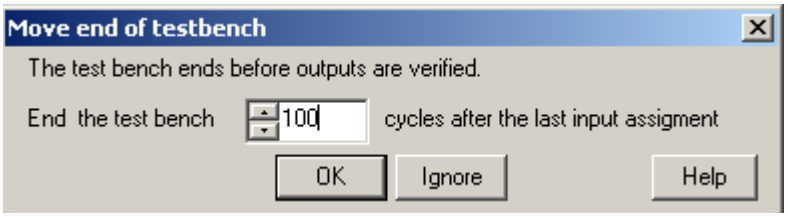


Рис. 16. Окно задания длительности тестовой последовательности

4-й этап. Получение конфигурационной последовательности передатчика байтов по стандарту RS232.

Для получения конфигурационной последовательности необходимо в окне *Sources in Project* выбрать файл *.vhd, а в окне *Processes for Current Source* двой-

ным щелчком мыши запустить *Generate Programming File*. В случае успеха в диалоговом окне появится сообщение *Done: completed successfully*.

Контрольные вопросы

1. Охарактеризовать стандарт RS232.
2. Перечислить режимы четности, дать их характеристику и предназначение.
3. Построить передатчик байтов по стандарту RS232 с числом бит в байте, равным 5-7 (по заданию преподавателя).
4. По заданию преподавателя реализовать фрагмент HDL-программы в схемотехническом редакторе.
5. По заданию преподавателя реализовать фрагмент схемы, построенной в схемотехническом редакторе, на языке VHDL.
6. Описать маршрут проектирования цифровых устройств на базе ПЛИС.

ЛАБОРАТОРНАЯ РАБОТА № 2

Разработка приемника байтов по стандарту RS232

Цель работы: изучение последовательного интерфейса RS232, построение приемника байтов по стандарту RS232 как конечного автомата с его последующей реализацией на языке VHDL.

Исходные данные заданы в табл. 1.

Задание:

1) построить автомат приемника байтов по стандарту RS232;
2) реализовать автомат приемника байтов на языке VHDL;
3) провести моделирование приемника с помощью HDL-Bencher;
4) получить конфигурационную последовательность приемника.

Пп. 1 – 2 задания необходимо выполнить при подготовке к работе.

Содержание отчета:

1) название, цель работы, задание;
2) схема автомата приемника байтов по стандарту RS232;
3) программа приемника байтов на языке VHDL;
4) результаты моделирования схемы.

Контрольные вопросы

1. Построить приемник байтов по стандарту RS232 с числом бит в байте равным 5-7 (по заданию преподавателя).
2. По заданию преподавателя реализовать фрагмент HDL-программы в схемотехническом редакторе.
3. По заданию преподавателя реализовать фрагмент схемы, построенной в схемотехническом редакторе, на языке VHDL.
4. Охарактеризовать микросхемы UART.

Библиографический список

1. Смит Дж. Сопряжение компьютеров с внешними устройствами. Уроки реализации: Пер с англ. – М.: Мир, 2000. – 266 с.
2. Ланцов В.Н. Проектирование ПЛИС на VHDL: Учеб. пособие / Владим. гос. ун-т. Владимир, 2000. 120 с.

3769

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

РЯЗАНСКАЯ ГОСУДАРСТВЕННАЯ РАДИОТЕХНИЧЕСКАЯ АКАДЕМИЯ

ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ УСТРОЙСТВ НА ОСНОВЕ САПР ФИРМЫ XILINX

Методические указания к лабораторным работам

Рязань 2005

УДК 621.315.6

Проектирование цифровых устройств на основе САПР фирмы Xilinx: Методические указания к лабораторным работам / Рязан. гос. радиотехн. акад.; Сост.: В.Н. Локтюхин, С.В. Челебаев, Н.В. Шемонаев. Рязань, 2005. 24 с.

Изложены краткие теоретические сведения и основы методов проектирования цифровых схем на основе ПЛИС фирмы Xilinx. Рассмотрено проектирование счетчиков, регистров и других устройств на основе ПЛИС.

Предназначены для студентов направления 200100.

Табл. 5 . Ил.17 . Библиогр.: 5 назв.

Проектирование, цифровые схемы, ПЛИС, Xilinx , САПР

Печатается по решению методического совета Рязанской государственной радиотехнической академии.

Рецензент: кафедра биомедицинской и полупроводниковой электроники РГРТА (зав. кафедрой С.В. Вихров)

Проектирование цифровых устройств на основе САПР фирмы Xilinx

Составители: Л о к т ю х и н Виктор Николаевич

Ч е л е б а е в Сергей Владимирович

Ш е м о н а е в Николай Викторович

Редактор Е.В. Ипатова

Корректор С.В. Макушина

Подписано в печать . Формат бумаги 60x84 1/16.

Бумага газетная. Печать трафаретная. Усл. печ. л. 1,5.

Уч.-изд. л. 1,5. Тираж 30 экз. Заказ

Рязанская государственная радиотехническая академия.

390005, Рязань, ул. Гагарина, 59/1.

Редакционно-издательский центр РГРТА.

ЛАБОРАТОРНАЯ РАБОТА № 1
ОСВОЕНИЕ ПАКЕТА ПРИКЛАДНЫХ ПРОГРАММ ФИРМЫ XILINX

Цель работы: ознакомиться с возможностями пакета прикладных программ фирмы Xilinx, научиться исследовать простые логические схемы.

1. Подготовка к работе

Перед началом работы следует ознакомиться с описанием, для этого необходимо изучить рекомендуемую литературу и приложение к данному пособию.

2. Теоретические сведения

2.1. Запуск нового проекта

- Запустите *Project Manager* двойным нажатием иконки (*Active-Cad*) в группе иконок *Active-Cad.*
- Когда появится окно *Project Manager*, нажмите иконку (*New Project*) для вызова окна *New Project*.
- Введите имя проекта, его каталог и тип в окне *New Project*. Некоторые типы проектов позволяют выбирать также семейство, номер детали и скорость устройства.
- Запустите схемный редактор нажатием кнопки *Schematic Editor* в окне *Project Manager.* Редактор откроет в своем окне новый схемный лист (*.sch*). Его имя будет тем же, что и имя текущего проекта.

2.2. Построение схемы

2.2.1. Рамка страницы и таблица

Схема может содержать много листов (страниц), которые могут создаваться командой *New Sheet* в меню *File.* Вы можете чертить схему на новой странице сразу после ее создания. Размер и рамка нового листа и его имя по умолчанию те же, что в текущем проекте. Они могут быть изменены с помощью опции *Page Setup* в меню *File*. Используя поля *Frame, Sheet Format* и *Ruler* вы можете определить:

- включение/выключение рамки и число зон на странице;
- размер листа (A-E, A0-A4, любой размер);
- ориентация листа (портрет или ландшафт);
- начало координатных осей.

Active-Cad позволяет заказывать таблицу проекта, которая отображается и печатается в правом нижнем углу листа схемы. Для заказа таблицы используйте опцию *Table Setup.*

2.2.2. Обозначения и компоненты

Каждое обозначение содержит графическую часть, имя компоненты, ее тип и атрибуты. Для поиска желаемого компонента нажмите любое

обозначение для вызова подсвечиваемого курсора. Далее пройдитесь по списку с помощью клавиш курсора или прямоугольника скроллинга. Если вы знаете имя обозначения, напечатайте его в строке редактирования внизу справа в окне *SC Symbols*. Подсвеченный курсор укажет на первое подходящее обозначение в списке. Для помещения на схеме нового элемента войдите в режим *Symbol* нажатием клавиши *F2* или иконки *Symbol Toolbox*. Обозначение из окна библиотеки автоматически появится на правой стороне экрана. Переместите обозначение, связанное с курсором мыши, в желаемую позицию и нажмите левую клавишу мыши. Когда вы выведите обозначение за пределы окна, схема будет автоматически прокручиваться под управлением средств авто-панорамирования. Для помещения другого компонента на схему выберите его в окне *SC Symbols* и повторите процедуру размещения компонента.

2.2.3. Соединения (провода или шины)

Проводные соединения могут быть начерчены:

- от контакта до контакта;
- от контакта до терминала ввода- вывода;
- от контакта к существующему проводному соединению и наоборот.

Вы можете закончить провод в любой позиции двойным нажатием кнопки мыши. Это произведет фиктивный терминал, также называемый концом провода (*wire-end*), который выглядит как кружек или маленький квадрат с красным крестиком внутри.

Все соединения действительны, пока они вычерчиваются, редактор не разрешает начинать или заканчивать провод, который не представляет действительного электрического соединения. Это ограничение - следствие подчиненной имитации системы *Active-Cad.*

2.2.4. Точки подключения питания

Активизируйте иконку *Power Symbol. Active-Cad* отобразит окно для выбора сигнала питания и его графического представления. Сделайте выбор и нажмите *Ok*. Курсор будет сопровождать маленький квадратик. Поместите его на желаемый контакт или провод и нажмите клавишу мыши. Произойдет подсоединение источника, если вместо *Ok* нажать *Repeat.*

2.2.5. Имена терминалов (клемм, зажимов) и схем

Имена схем используются для наименования проводов (узлов) на схеме. Все схемы, использующие одинаковые имена, объединяются в одну. Вначале проводам присваиваются скрытые имена. Они используются внутри *Active-Cad* для прокладки проводов. Однако для пояснения и документирования вы можете дать имя каждому проводу на схеме.

Для добавления имени к существующему проводу используется следующая процедура

- Нажмите иконку *Add Net or Bus Name.*
- Когда появится окно *Net Name,* введите имя и нажмите *Ok.*
- При курсоре появится красный прямоугольник. Переместите его в позицию, где должно быть помещено имя, и нажмите клавишу мыши. Имя схемы будет помещено около провода. С другой стороны, оно отобразится другим цветом (зеленым), в отличие от имени схемы (синий), и не будет представлено проводу. Изменение цвета в меню *Preference* изменит ранее описанные цвета.

Примечание. Каждый узел может иметь только одно имя. Если вы попытаетесь соединить две схемы с одинаковыми именами, на схеме появится предупреждение. Вы должны изменить или удалить одно из имен перед соединением двух схем.

Если вы попытаетесь присвоить новое имя уже поименованному проводу, будет выдано предупреждение, а операция отменена. Вы можете присвоить новое имя только после удаления первоначального.

Для изменения имени дважды нажмите имя схемы и, когда появится окно *Net Name,* редактируйте имя.

Для удаления имени схемы выберите имя и нажмите клавишу *Delete.*

Для передвижения имени схемы включите режим *Select*, нажмите имя схемы и переместите его в желаемую позицию.

2.2.6. Графика и текст

Графический редактор выбирается иконкой *Grafics Toolbox* или клавишей *F6*. Вычерчиваемые элементы не имеют электрических параметров и не проверяются по электрическим правилам. Опции графического редактора:

- *Line* - чертит единственную линию;
- *Polyline* - чертит многосегментные линии;
- *Ortholine* - чертит только горизонтальные или вертикальные линии;
- *Rectangle* - чертит прямоугольник по выбранным диагональным углам;
- *Circle* - чертит круг с выбранным центром и вытягиванием (*stretching*) его внешней линии;
- *Arc* - чертит дугу путем выбора конечных точек и проведением внешней линии;
- *Text* - создание и размещение строк текста на схеме;
- *Text Style* - установка размера текста, шрифта и цвета;
- *Line Style* - установка формы линии, цвета и толщины.

Каждая операция редактирования графики начинается и заканчивается нажатием левой клавиши мыши. Вы можете приостановить текущий режим нажатием правой клавиши мыши с последующим выбором иконки *End* в панели инструментов *SC Graphics*.

2.3. Внесение изменений в схему

2.3.1. Выбор области схемы

Для просмотра специфичной области схемы используйте опции:

- увеличить выбранную область;
- увеличить полный лист схемы;
- обменять текущую страницу с изображением полной страницы;
- поместить выбранную область в центр изображения схемы.

2.3.2. Увеличение в выбранной области

Выберите иконку *Zoom Area* в верхнем прямоугольнике и переместите ее в угол, противоположный углу выбранной области. При освобождении клавиши мыши отмеченная область распространится на весь экран.

Для увеличения внутри выбранной области нажмите на один из углов увеличиваемой области и нажмите одновременно клавишу мыши и *CTRL*. Удерживая клавишу мыши нажатой, переместите курсор в противоположный угол увеличиваемой области. Отмеченный прямоугольник отображается во время операции для отметки выбранной области. Когда желаемая область окажется внутри прямоугольника, освободите клавишу мыши. Выбранная область расширится на весь экран.

Для центрирования выбранной области на изображении схемы необходимо нажать *F9* или выбрать опцию *Center* в меню *Display* . В ответ *Active-Cad* отобразит курсор в виде прицела. Если вы нажмете левую клавишу мыши, область под курсором отобразится в центре экрана.

2.3.3. Дублирование обозначений

При активной иконке *Symbol Toolbox* нажатие на любое обозначение производит его копию (дубликат), который следует за курсором мыши. Вы можете поместить дубликат в любую позицию экрана нажатием на клавишу мыши. Это разрешено, если схема находится в только режиме *Symbol* (размещение), и никакое другое обозначение не представлено курсору. Если обозначение представлено курсору, нажмите правую клавишу мыши для освобождения существующего обозначения, а затем нажмите на желаемое обозначение, которое вы хотите скопировать. Каждое помещенное обозначение получит свой ссылочный номер.

2.3.4. Нумерация обозначений

Все обозначения автоматически именуются при помещении в схему. Имя состоит из текстовой строки и числового суффикса, который

увеличивается каждый раз, когда вы помещаете обозначение. Для переименования детали выберите опцию *Annotation* в меню *Options.*

2.3.5. Поворот (вращение) обозначений

Для вращения по часовой стрелке отдельного обозначения нажмите клавиши *Ctrl+L*, пока обозначение представлено курсору. Каждое нажатие поворачивает обозначение на 90^0. Нажатие *Ctrl+M* порождает зеркальное отражение обозначения.

Если обозначение уже помещено на схеме, подсветите его в режиме *Select* и нажмите иконку *Properties*. Это вызовет окно *Edit Symbol.* В окне *Edit Symbol* выберите опцию *Attributes,* выберите желаемый поворот компоненты.

2.3.6. Удаление обозначений

Для удаления обозначений, только что введенных, нажмите клавишу *Del* на панели. По каждому нажатию удаляется по одному обозначению в порядке, обратном их вводу в схему. Операция удаления запрещена только, когда вы выходите из режима *Symbol Toolbox*. Возврат обратно в этот режим не восстанавливает опцию удаления.

Вы можете удалить любое обозначение, терминал или провод в режиме *Select.* Для удаления элемента, выберите его нажатием клавиши мыши и тогда нажмите клавишу *Del* на панели или воспользуйтесь опцией *Delete* в меню *Edit.* Если вы ошибочно удалили обозначение, можете восстановить его нажатием опции *Undo* в меню *Edit.*

2.3.7. Сохранение проекта

Для того чтобы сохранить проект необходимо выбрать опцию File: Save (комбинация клавиш Ctrl-S) или нажать соответствующую иконку на горизонтальной панели редактора.

3. Порядок выполнения работы

1. Получить у преподавателя вариант задания согласно номеру бригады (см. таблицу), провести минимизацию функции.
2. Сформировать логическую схему.
3. Сформировать полученную логическую схему с помощью пакета прикладных программ.
4. Представить отчет о работе: распечатка логической схемы и результатов моделирования.

Варианты заданий к лабораторной работе № 1

№	Варианты заданий
1	$f = x_1\bar{x}_2x_3x_4 + x_1\bar{x}_3 + x_2x_4$
2	$f = x_1x_2x_4 + x_2\bar{x}_3 + \bar{x}_1\bar{x}_2\bar{x}_3$
3	$f = \bar{x}_1x_2 + x_3x_4 + \bar{x}_2\bar{x}_3x_4$
4	$f = x_1\bar{x}_3 + \bar{x}_3x_4 + \bar{x}_1\bar{x}_2x_3$

5	$f = \bar{x}_1 x_3 + \bar{x}_1 x_2 \bar{x}_3 + x_2 \bar{x}_4$
6	$f = \bar{x}_1 x_2 x_3 + x_1 \bar{x}_3 + \bar{x}_2 x_3 \overline{x_4}$
7	$f = \bar{x}_1 \bar{x}_3 x_4 + x_2 x_3 + x_1 + \bar{x}_2 x_4$
8	$f = \bar{x}_2 \bar{x}_3 + \bar{x}_1 x_3 x_4 + x_1 x_2 \overline{x_4}$
9	$f = x_1 x_2 \bar{x}_3 + \bar{x}_1 x_2 x_4 + x_3 \overline{x_4} x_2$
10	$f = \bar{x}_1 x_2 \overline{x_4} + \bar{x}_1 \bar{x}_2 \bar{x}_3 + x_2 \overline{x_4}$

ЛАБОРАТОРНАЯ РАБОТА № 2

МОДЕЛИРОВАНИЕ КОМБИНАЦИОННЫХ ЦИФРОВЫХ УСТРОЙСТВ

Цель работы: обучение методике проектирования комбинационных цифровых устройств, закрепление навыков применения ППП фирмы Xilinx.

1. Теоретические сведения

Перед началом работы рекомендуется изучить теорию, представленную в рекомендуемой литературе по принципам синтеза комбинационных схем.

Методика структурного проектирования комбинационных схем

Комбинационными схемами называются цифровые устройства, логическое состояние выходов которых зависит только от комбинации логических сигналов на входах в данный момент времени.

Исходными данными для проектирования комбинационной схемы являются его функциональное описание и требования к основным электрическим параметрам. Функциональное описание комбинационной схемы обычно дается в виде таблицы истинности или алгебраического выражения. На стадии структурного проектирования получается наиболее простое (минимизированное) логическое выражение заданной функции и строится соответствующая ему структурная схема на заданной или выбранной элементной базе. Процесс структурного проектирования разбивается на несколько последовательно выполняемых этапов.

1. Минимизация заданной логической функции.

Чем проще логическое выражение реализуемой функции, тем меньше элементов требуется для ее выполнения. В результате микросхема, выполняющая эту функцию, будет иметь лучшие показатели по быстродействию, потребляемой мощности, площади кристалла.

В алгебре логики разработаны методы минимизации логических функций. Для минимизации функций относительно небольшого числа переменных ($k \leq 6$) наиболее простым и наглядным является графический

метод, использующий карты Карно. Для функций большего числа переменных (k>6) можно декомпозицией выделить более простые составляющие функции с числом переменных не более 6, которые минимизируются с помощью карт Карно.

При использовании этого метода исходная логическая (или переключательная) функция представляется на карте Карно. Минтермы, соответствующие двум соседним (в столбце или ряду) клеткам карты Карно, отличаются значениями только одной переменной. Поэтому дизъюнкция этих двух минтермов дает один конъюнктивный член, в котором исключена общая переменная [3].

Чтобы выделить клетки, представляющие один конъюнктивный член, их графически объединяют — обводят на карте Карно тонкой линией. Таким образом, конъюнктивный член представляется на картах Карно двумя объединенными клетками. Такое объединение клеток на картах Карно эквивалентно выполнению данного ранее логического преобразования и позволяет получить более простое выражение логической функции. Объединенные клетки будут соответствовать конъюнктивным членам, дизъюнкция которых даст минимизированную дизъюнктивную нормальную форму (МДНФ) заданной функции.

2. Построение структурной схемы проектируемой микросхемы в соответствии с полученными МДНФ или МКНФ

При этом полученные логические выражения сначала следует представить в виде комбинации операций, выполняемых элементами базового набора, а затем построить логическую структуру схемы. Базовые элементы современных цифровых микросхем выполняют операции И — НЕ, ИЛИ — НЕ, И — ИЛИ — НЕ, НЕ и Монтажное И ($И^2Л$), ИЛИ — НЕ и Монтажное ИЛИ (ЭСЛ).

Примеры типичных комбинационных схем

1. Шифраторы и дешифраторы

Одними из основных видов преобразования информации в цифровых системах является *шифрация и дешифрация.* Шифрацией называется преобразование M-разрядного двоичного кода, имеющего $К_м$ безразличных наборов входных переменных, в однозначно соответствующий ему n-разрядный код, имеющий меньшее число разрядов $n < M$ и безразличных наборов $К_n < К_м$. Таким образом, при шифрации каждому из $(2^м — Км)$ рабочих наборов входных переменных ставится в соответствие один из $(2^n — К_n)$ рабочих наборов выходных переменных, т. е. $(2^м — К_м) = (2^n — К_n)$. В результате шифрации осуществляется «сжатие» информации для передачи но меньшему числу линий связи ($n<M$) за счет полного (при $К_n = 0$) или частичного (при $К_n \neq 0$) исключения безразличных наборов. Обратное

преобразование, т. е. восстановление информации в первоначальном *M*-разрядном коде с $K_м$ избыточными комбинациями, называется дешифрацией. Функциональные узлы для выполнения этих операций называются *шифраторами* и *дешифраторами.* Они различаются по числу входов и выходов и называются «шифратор (дешифратор) из *M* в *n (*из *n* в *M)*». Максимальное число входов шифратора не превышает количество возможных комбинаций выходных сигналов: $M < 2^m$. Соответственно для дешифратора число выходов $n \leq 2^m$ [3]. Дешифратор реализует на выходах минтермы входных переменных или инверсии минтермов — макстермы.

2. Мультиплексоры и демультиплексоры

Мультиплексором называется функциональный узел, обеспечивающий передачу информации, поступающей по нескольким входным линиям связи, на одну выходную линию. Выбор той или иной входной линии A_i осуществляется в соответствии с поступающим адресным кодом S_0S_1... При наличии *n* адресных входов можно реализовать $M = 2^n$ комбинаций адресных сигналов S_i, каждая из которых обеспечивает выбор одной из *M* входных линий. Чаще всего используются мультиплексоры «из 4 в 1» *(M* = 4, *n* = 2), «из 8 в 1» *(M* = 4, *n* = 3), «из 16 в 1» *(M.* = 16, *n* = 4). Если допустить наличие избыточных комбинаций адресных сигналов, то можно спроектировать мультиплексоры с любым числом входных линий $M \leq 2^n$.

Для восстановления мультиплексированной информации используются *демультиплексоры,* которые в соответствии с принятым адресом направляют информацию в одну из *M* выходных линий связи. При этом на остальных линиях поддерживается логический "0".

Мультиплексоры и демультиплексоры могут быть реализованы на элементах различных типов путем выполнения логических выражений.

2. Порядок выполнения работы

2.1. Получить у преподавателя вариант задания согласно номеру бригады (см. таблицу).

2.2. В соответствии с заданием сформировать таблицу истинности требуемого устройства.

2.3. На основании этой таблицы получить необходимые СДНФ.

2.4. С помощью карт Карно провести минимизацию полученных СДНФ.

2.5. На основании полученных в результате минимизации МДНФ, построить структурную схему устройства.

2.6. Сформировать полученную схему с помощью ППП.

2.7. Провести исследование заданной схемы и убедиться в правильности ее работы.

2.8. Представить отчет: распечатка логической схемы и результатов моделирования.

Варианты заданий к лабораторной работе № 2

№	Варианты заданий	
	Логическая схема	**Базис**
1	Шифратор "из 10 в 4"	И-НЕ
2	Шифратор "из 10 в 4"	ИЛИ-НЕ
3	Дешифратор "из 4 в 10"	И-НЕ
4	Дешифратор "из 4 в 10"	ИЛИ-НЕ
5	Мультиплексор "8 на 1"	И-НЕ
6	Мультиплексор "8 на 1"	ИЛИ-НЕ
7	Демультиплексор "1 на 8"	И-НЕ
8	Демультиплексор "1 на 8"	ИЛИ-НЕ
9	5- разрядный мажоритарный элемент	И-НЕ
10	5- разрядный мажоритарный элемент	ИЛИ-НЕ

Библиографический список

1. Алексенко А.Г., Шагурин И.И. Микросхемотехника: Учеб. пособие для вузов. 2-е изд., перераб. и доп. М.: Радио и связь, 1990. С. 156 - 237.
2. Цифровая и вычислительная техника: Учебник для вузов / Э.В. Евреинов, Ю.Т. Бутыльский, И.А. Мамзелев и др.; Под ред. Э.В. Евреинова М.: Радио и связь, 1991. С. 91 - 147.
3. Потемкин И.С. Функциональные узлы цифровой автоматики. М.: Энергоатомиздат, 1988. С. 121 - 149.

ЛАБОРАТОРНАЯ РАБОТА № 3
МОДЕЛИРОВАНИЕ ПОСЛЕДОВАТЕЛЬНОСТНЫХ ЦИФРОВЫХ УСТРОЙСТВ

Цель работы: обучение методике проектирования последовательностных цифровых устройств, изучение дополнительных возможностей пакета прикладных программ для проектирования ПЛИС фирмы XILINX .

1. Подготовка к работе

Перед началом работы требуется изучить теорию, представленную в рекомендуемой литературе.

2. Теоретические сведения

Основными типами последовательностных функциональных узлов являются регистры, счетчики и генераторы чисел.

В данной работе рассмотрены счетчики. *Счетчиком* называется узел, на выходах которого образуется число, соответствующее количеству поступивших на вход импульсов. Основным параметром счетчика является

модуль счета K_c. — максимальное число импульсов, которое может быть сосчитано счетчиком. После поступления K_c импульсов счетчик должен возвращаться в исходное состояние. Величина K_c равна числу устойчивых состояний счетчика. Счетчик, содержащий m разрядов (триггеров), может иметь 2^m устойчивых состояний, поэтому его модуль счета $K_c \leq 2^m$. Количество поступивших на счетный вход импульсов представляется на выходе счетчика в виде двоичного числа в том или ином коде: прямом дополнительном, циклическом. Обычно счетчики имеют дополнительные входы установки *S*, позволяющие предварительно установить на выходе счетчика заданное число, или входы сброса *R,* сигнал на которых (по переднему или заднему фронту) переводит счетчик в исходное состояние [3].Частота импульсов на выходе последнего разряда счетчика в K_c раз меньше, чем частота импульсов, поступающих на вход. Поэтому счетчики используются в качестве *делителей частоты,* обеспечивающих на выходе в K_c раз меньшую частоту сигнала, чем на входе.

По типу функционирования различают счетчики суммирующие, вычитающие и реверсивные. *Суммирующий* счетчик выполняет прямой счет, т. е. при поступлении на вход очередного импульса число на выходе счетчика увеличивается па единицу. *Вычитающий* счетчик производит обратный счет, т. е. при поступлении счетного импульса число на выходе уменьшается на единицу. *Реверсивный* счетчик может работать в режимах прямого и обратного счета.

Счетчики широко используются в устройствах управления цифровых систем для подсчета числа выполненных операций, в связной и контрольно-измерительной аппаратуре, для определения числа поступивших сигналов и уменьшения их частоты и др. По структурной организации счетчики делятся на параллельные, последовательные и параллельно-последовательные. Они различаются способами подачи счетных импульсов на входы разрядов.

Параллельные счетчики. В счетчиках этого типа счетные импульсы одновременно (параллельно) поступают на синхровходы С триггеров во всех разрядах. Параллельные счетчики обычно строятся на базе *T-, JK-, D-* триггеров, синхронизируемых фронтом.

Структурное проектирование таких счетчиков выполняется с помощью словарного метода [1]. При увеличении модуля счета, т. е. числа разрядов m, в параллельных счетчиках возрастают количество логических элементов и триггеров, требования к числу их входов, существенно усложняется система межсоединений. Поэтому параллельные счетчики $m > 6$ — 8 используются редко. Для получения модулей $K_c > 16$ обычно применяются последовательные или параллельно-последовательные счетчики [3].

Последовательные и параллельно-последовательные счетчики. В счетчиках этого типа счетные импульсы подаются только на вход триггера первого разряда. Для каждого из последующих разрядов сигналы переключения поступают с выхода предыдущих разрядов. В результате происходит последовательное переключение разрядов счетчика.

Последовательные счетчики с модулями счета K_c обычно реализуются последовательным соединением m Т-триггеров, каждый из которых работает как счетчик с $K_c = 2$. Структурная схема и временная диаграмма такого суммирующего счетчика с $K_c = 16$ показаны на рис.1.

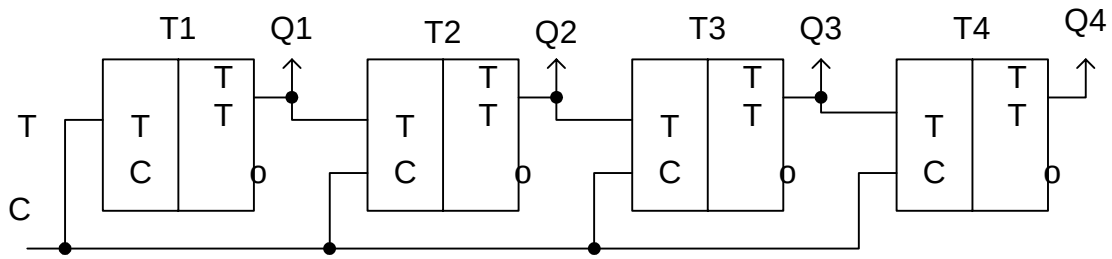


Рис.1. Последовательный счетчик на Т - триггерах

Отметим, что данная схема будет работать как вычитающий счетчик, если вход T каждого последующего (k-го) триггера подключить к неинвертирующему выходу Q_{k-1} предыдущего триггера, а асинхронные входы S_d использовать для установки какого-либо начального числа. Реверсивный счетчик можно реализовать, включив на счетном входе каждого из триггеров элемент И—ИЛИ—НЕ, выполняющий функцию

$$Tk = \overline{MQk+1+\overline{M}Qk-1} \quad .$$

Для получения модулей счета $K_c \neq 2^m$ в последовательных счетчиках на Т-триггерах требуется включение дополнительных логических элементов. Один из способов реализации таких счетчиков состоит в использовании входов сброса триггеров. Например, декадный счетчик (K_c = 10) можно получить из последовательного 4-разрядного счетчика с K_c = 16, если при поступлении 10-го импульса T = 1, когда установится состояние выходов $Q_0Q_1Q_2Q_3$ = 0101, подать на общий и ход сброса сигнал R = 1. Для этого используется управляющая комбинационная схема, выходная функция R которой равна 1 только при $Q_3 = Q_1 = 1$ и $Q_0 = Q_2 = 0$. После минимизации получаем выражение функции

$$R = \overline{\overline{Q}1+\overline{Q}3},$$

которое реализуется с помощью элемента ИЛИ — НЕ, подключаемого ко входу R последовательного счетчика.

Обозначения триггеров в программе:

FDC – D-триггер;

FTC – T-триггер;

FJKC – JK-триггер;
FJKRSE – JK- триггер с преимущественным сбросом;
FJKSRE – JK-триггер с преимущественной предустановкой.

2. Порядок выполнения работы

1. Получить у преподавателя вариант задания согласно номеру бригады (см. таблицу).
2. Спроектировать логическую схему последовательностного устройства по методике, изложенной в [1].
   - Определить количество ячеек (триггеров), необходимых для реализации заданного устройства.
   - Составить таблицу состояний устройства.
   - На основании этой таблицы составить карты Карно для функций переходов.
   - Минимизировать функции входов триггеров.
   - Сформировать логическую схему устройства.
3. Сформировать полученную схему с помощью пакета прикладных программ фирмы XILINX, руководствуясь указаниями к лабораторной работе № 1 и используя логические элементы и триггеры из стандартной библиотеки.
4. Провести исследование заданной схемы и убедиться в правильности ее работы.
5. Представить отчет: распечатка логической схемы и результатов моделирования.

Библиографический список

1. Основы синтеза логических схем: Учеб. пособие / В.Н. Локтюхин. Рязан. гос. радиотехн. академ. Рязань, 2004. 48 с.
2. Цифровая и вычислительная техника: Учебник для вузов / Э.В. Евреинов, Ю.Т. Бутыльский, И.А. Мамзелев и др.; Под ред. Э.В. Евреинова М.: Радио и связь, 1991. С. 91 - 147.
4. Потемкин И.С. Функциональные узлы цифровой автоматики. М.: Энергоатомиздат, 1988. С. 121 - 149.

Варианты заданий к лабораторной работе № 3

№	Тип счетчика	Коэффициент счета	Тип триггеров
1	Параллельный суммирующий	10	D
2	Параллельный вычитающий	11	D
3	Параллельный суммирующий	11	T

4	Параллельный вычитающий	12	T
5	Параллельный суммирующий	10	JK
6	Параллельный вычитающий	11	JK
7	Параллельный суммирующий	12	D
8	Параллельный вычитающий	13	D
9	Параллельный суммирующий	14	T
10	Параллельный вычитающий	13	T
11	Параллельный суммирующий	12	JK
12	Последовательный суммирующий	11	JK
13	Последовательный вычитающий	11	D
14	Последовательный суммирующий	12	D
15	Последовательный вычитающий	12	JK

ЛАБОРАТОРНАЯ РАБОТА № 4

РАЗРАБОТКА СХЕМЫ ФОРМИРОВАНИЯ ШИРОТНО-ИМПУЛЬСНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ

Цель работы: обучение методики проектирования схем формирования широтно-импульсной последовательности.

1. Подготовка к работе

Перед началом работы требуется изучить теорию, представленную далее.

2. Теоретические сведения

Широтно-импульсная модуляция (ШИМ) характеризуется относительной длительностью (скважностью) $Q = \tau / T$, где T – период импульсного сигнала, τ – длительность единичного фронта (рис.1).

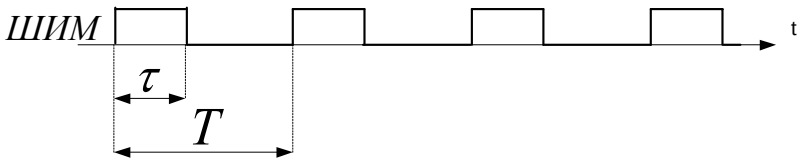


Рис.1. Модулированный широтно-импульсный сигнал

Скважность может быть как одинаковой, так и переменной. Для построения схемы, формирующей ШИМ с постоянной скважностью (рис.2), требуется счетчик и два компаратора. Счетчик, работающий по тактовой частоте *clk*, формирует длительность периода T, по достижении которого сбрасывается в начальное значение. Сигнал сброса *reset* формирует первый компаратор. Второй компаратор сравнивает текущее значение *count* счетчика с величиной τ внутри периода T. Выходной сигнал схемы *shim* снимается с выхода второго компаратора.

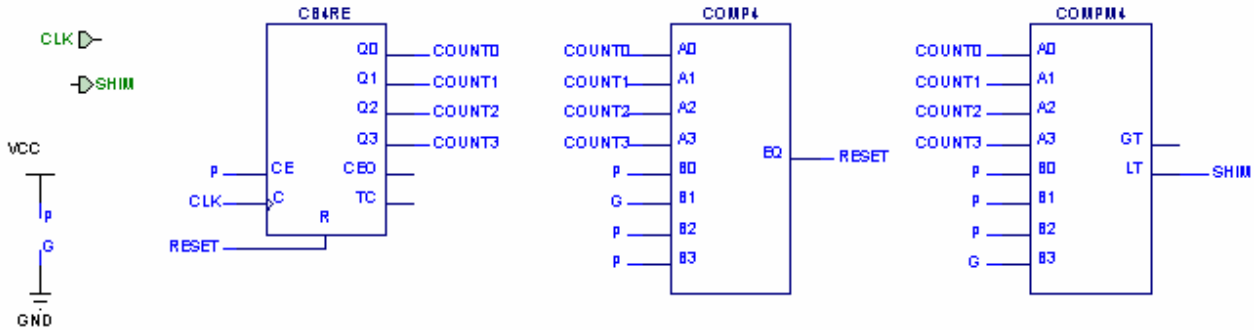


Рис.2. Схема формирования ШИМ с постоянной скважностью $Q = 7/14$

Временные диаграммы работы схемы, приведенной на рис. 2.2, приведены далее (рис. 2.3):

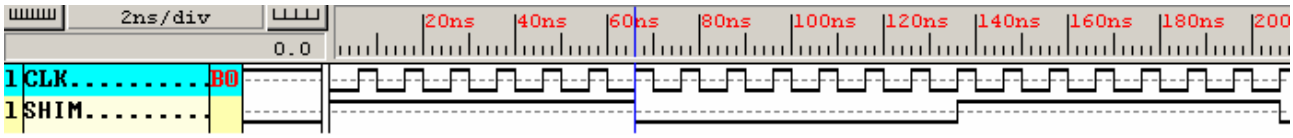


Рис.3. Результаты моделирования схемы формирования ШИМ с постоянной скважностью $Q = 7/14$

Схема формирования ШИМ с переменной скважностью (рис.4) состоит из двух счетчиков и трех компараторов. Первый счетчик формирует сигнал *count*, отсчитывающий период T. Сброс его осуществляется либо по сигналу запуска *start*, либо по достижении периода T. Сигнал *reset* вырабатывается первым компаратором как признак окончания периода T. По сигналу *reset* происходит также приращение скважности на втором счетчике. Выходной сигнал схемы *shim* снимается с выхода второго компаратора. Третий компаратор вырабатывает признак очередного повторения цикла *reset_cycle*.

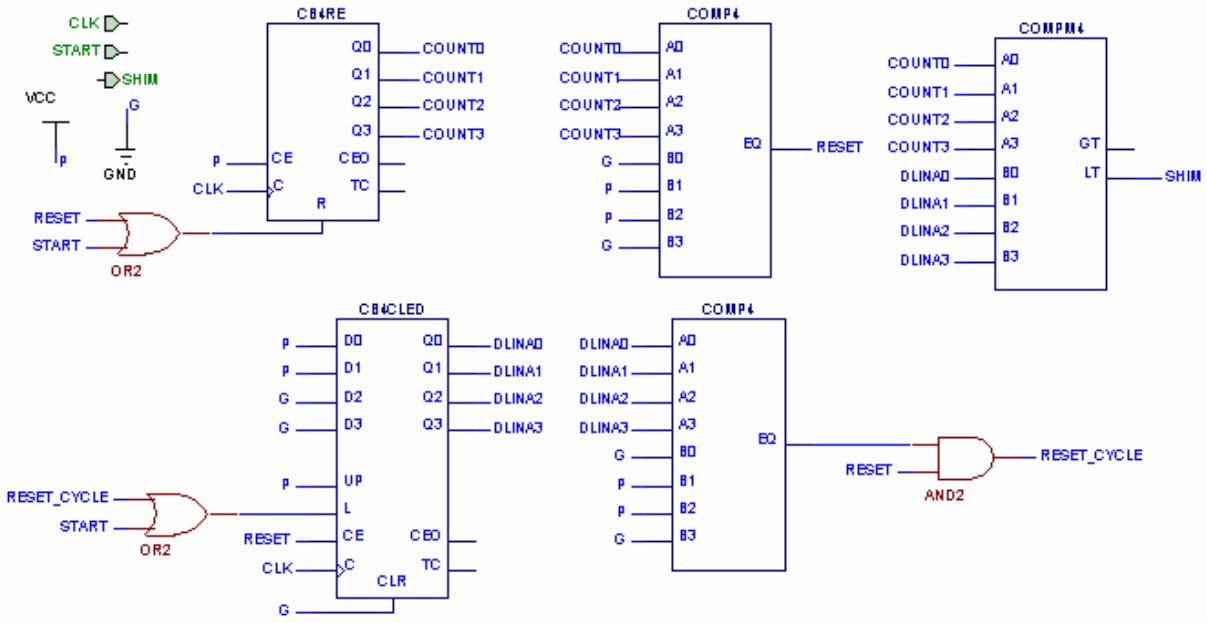


Рис. 4. Схема формирования ШИМ с переменной скважностью $Q = 3/7 \div 6/7$

Временные диаграммы работы схемы, приведенной на рис. 4, приведены далее (рис.5):

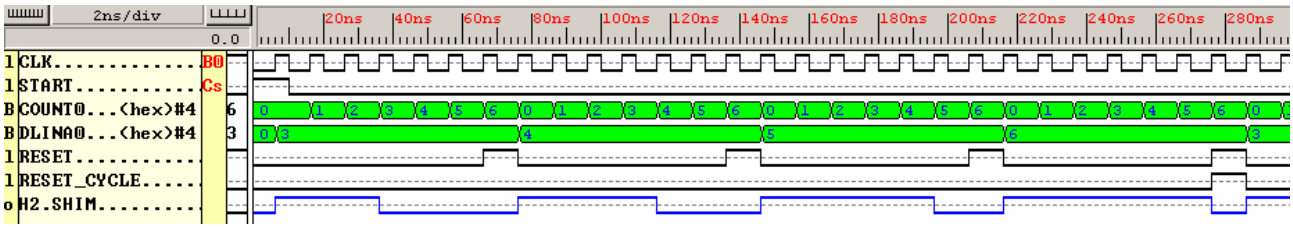


Рис. 5. Результаты моделирования схемы формирования ШИМ с переменной скважностью $Q = 3/14 \div 6/14$

3. Порядок выполнения работы

3.1. Получить у преподавателя задание согласно номеру бригады (см. таблицу).
3.2. Провести синтез счетчика требуемого типа (см. лаб. Работу №3).
3.3. Построить схему формирования ШИМ.
3.4. Составить отчет.

Варианты заданий к лабораторной работе № 5

	Вид скважности	$Q = \tau / T$	Тип счетчика	Разрядность
1	Постоянная	5/15	D	4
2	Постоянная	6/11	T	4
3	Постоянная	3/13	D	4

4	Постоянная	9/11	T	4
5	Постоянная	4/12	D	4
6	Постоянная	8/13	T	4
7	Постоянная	11/15	D	4
8	Постоянная	4/15	T	4
9	Переменная	1/7 – 6/7	D	3
10	Переменная	1/7 – 6/7	T	3
11	Переменная	2/7 – 5/7	D	3
12	Переменная	2/7 – 5/7	T	3
13	Переменная	1/7 – 5/7	D	3
14	Переменная	1/7 – 5/7	T	3
15	Переменная	2/7 – 6/7	D	3
16	Переменная	2/7 – 6/7	T	3

ЛАБОРАТОРНАЯ РАБОТА № 5
РАЗРАБОТКА ДИСПЕТЧЕРА ПАМЯТИ

Цель работы: обучение методике проектирования диспетчера памяти, изучение принципов работы с внешним запоминающим устройством.

1. Подготовка к работе

Перед началом работы требуется изучить теорию, представленную далее.

2. Теоретические сведения

Диспетчер памяти необходим для обращения к одному модулю памяти от нескольких потребителей (каналов). Диспетчер памяти состоит из двух компонент: **контроллера памяти** и **блока приоритетов**.

Контроллер памяти осуществляет:

1) выдачу сигналов разрешения записи и чтения на память;
2) установку адреса обращения к памяти;
3) переключение двунаправленной шины данных в зависимости от вида обращения (записи или чтения);
4) прием данных из памяти при чтении.

Блок приоритетов формирует ответ тому каналу, запрос на обращение к памяти по которому следует удовлетворить. По алгоритму функционирования блоки приоритетов делятся на абсолютные и относительные.

В блоке с абсолютными приоритетами все каналы имеют свой строго определенный приоритет. Это означает, что если имеется запрос по каналу с наивысшим приоритетом, то выполняется именно он. Запрос по каналу с

самым низким приоритетом выполняется только в том случае, если нет запросов ни по одному из других каналов.

В блоке с относительными приоритетами ответ выдается тому каналу, в очереди запросов которого находится наибольшее количество элементов.

На рис.1 показан двухканальный блок абсолютных приоритетов, где *req_a* – запрос по каналу с высшим приоритетом, *req_b* – запрос по каналу с низшим приоритетом, *ask_a* и *ask_b* – ответы на запросы по каналам, *ask* – общий ответ. В следующем такте после выставления сигнала *ask* на контроллер памяти подаются признак чтения/записи *rd_wr* (*rd_wr*=1 при чтении), адрес *adr* и данные *d_in* (при записи).

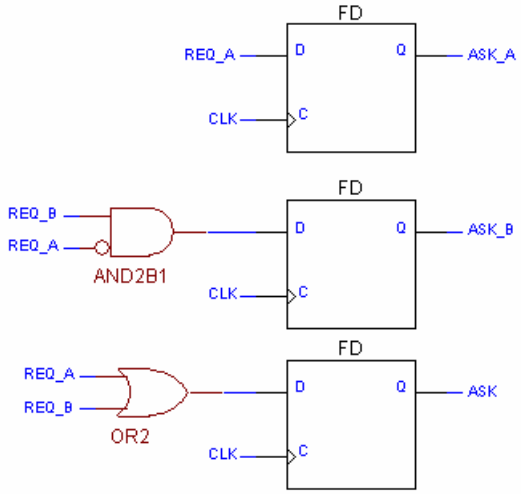


Рис.1. Двухканальный блок абсолютных приоритетов

На рис. 2 показан двухканальный блок относительных приоритетов. На схеме имеют место следующие обозначения: *cnt_a* – число неотвеченных запросов по каналу *a*, *cnt_b* – число неотвеченных запросов по каналу *b*.

Функционирование контроллера памяти зависит от ее типа. Память бывает синхронной и асинхронной.

Асинхронная память характеризуется длительностью цикла $t_ц$ обращения к ней. Это время, как правило, составляет 20-30 ns. Особенностью данного типа памяти является то, что между соседними обращениями на запись и чтение должен быть свободный цикл. Временные диаграммы по чтению и записи асинхронной памяти показаны на рис.3.

Для асинхронной памяти максимальная тактовая частота *CLK* работы схемы выбирается исходя из выражения: $CLK \le 2/t_ц$. При удовлетворении данного условия цикл обращения по записи или чтению длится 2 такта.

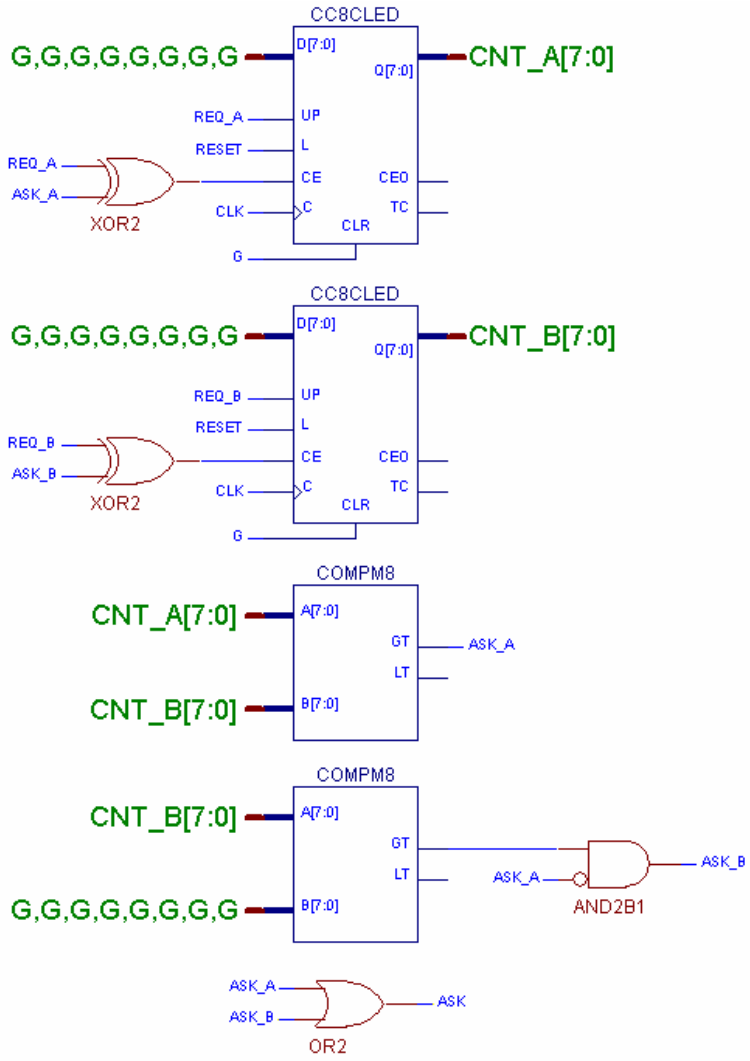


Рис.2. Двухканальный блок относительных приоритетов

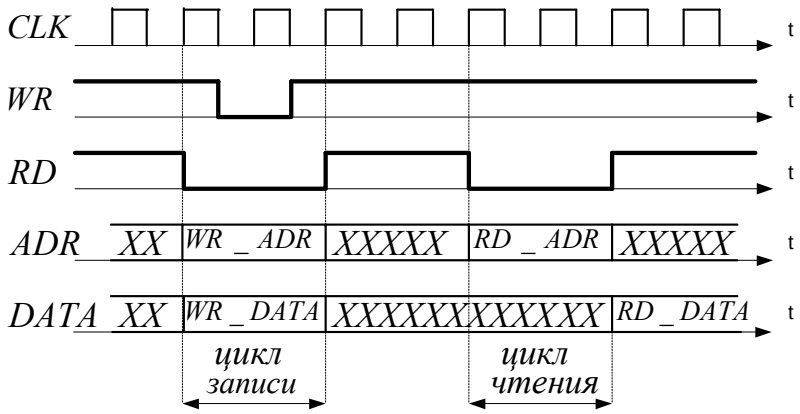


Рис. 3. Временные диаграммы циклов чтения и записи асинхронной памяти

Сигналы разрешения записи WR и чтения RD при отсутствии обращения к памяти устанавливаются в "1". При чтении памяти сигнал RD устанавливается в "0" и удерживается в течении времени $t_ц$. При этом на память выставляется адрес $ADR = RD_ADR$. По истечении $t_ц$ с памяти снимаются данные RD_DATA. При записи в память сигнал RD

устанавливается в "0". Вместе с этим на память выставляются адрес WR_ADR и данные WR_DATA. Сигнал WR устанавливается в "0" по истечении времени, равного $t_ц / 4$ после начала обращения, и удерживается в данном положении в течении интервала времени, равного $t_ц / 2$. Схема формирования сигналов *WR* и *RD* приведена на рис. 1.4.

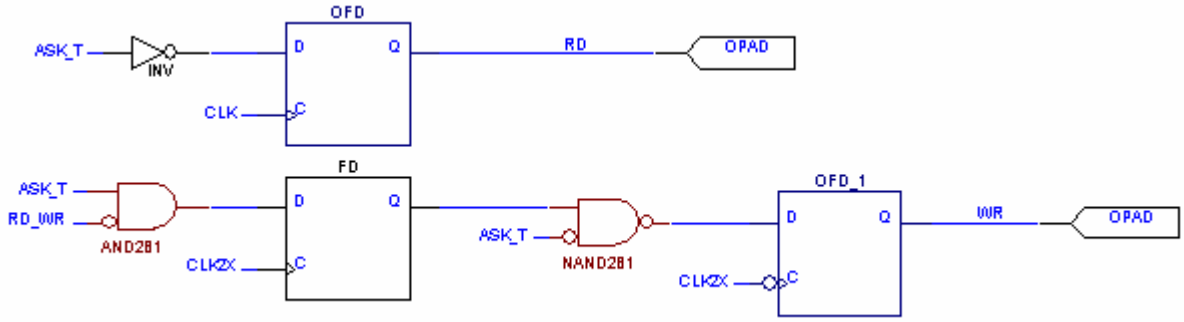


Рис.4. Установка признаков чтения и записи для асинхронной памяти

Синхронная память является тактируемой. Цикл обращения к ней составляет 2-3 такта. Длительность обращения здесь определяется периодом тактовой частоты, помноженным на число тактов цикла обращения. Временные диаграммы по чтению и записи синхронной памятью с 2-тактовой задержкой показаны на рис. 1.5.

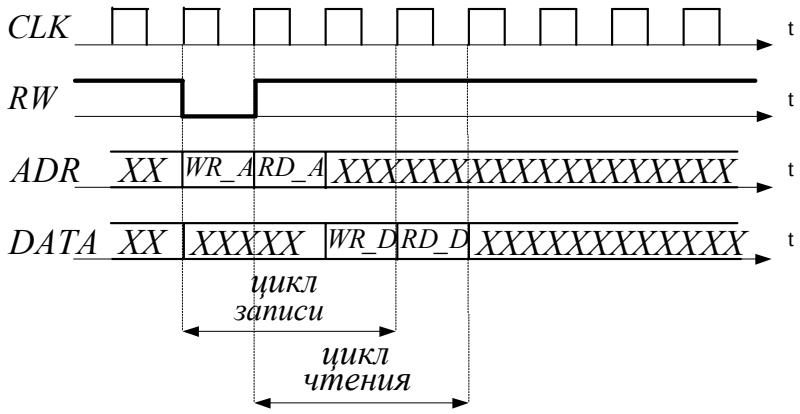


Рис. 5. Временные диаграммы циклов чтения и записи асинхронной памяти

Сигнал признака чтения/записи RW при отсутствии обращения к памяти устанавливается в "1". При чтении сигнал RW устанавливается в "1" и удерживается один такт. При этом на память выставляется адрес $ADR = RD_A$. Спустя 2 такта с памяти снимаются данные RD_D. При записи сигнал RW устанавливается в "0". Вместе с этим на память выставляются адрес WR_A. Данные WR_D подаются на память спустя 2 такта.

Установка адреса обращения памяти показана на рис.6:

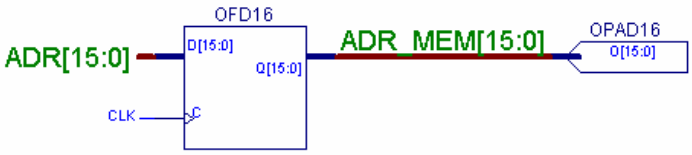


Рис. 6. Установка адреса обращения к синхронной и асинхронной памяти

Схема выдачи признака чтения/записи синхронной памяти показана на рис. 7:

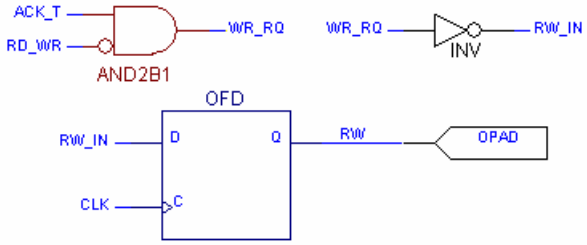


Рис. 7. Выдача признака чтения/записи на синхронную память

Переключение двунаправленной шины данных в зависимости от вида обращения для синхронной и асинхронной памяти показано на рис.8 и 9 соответственно.

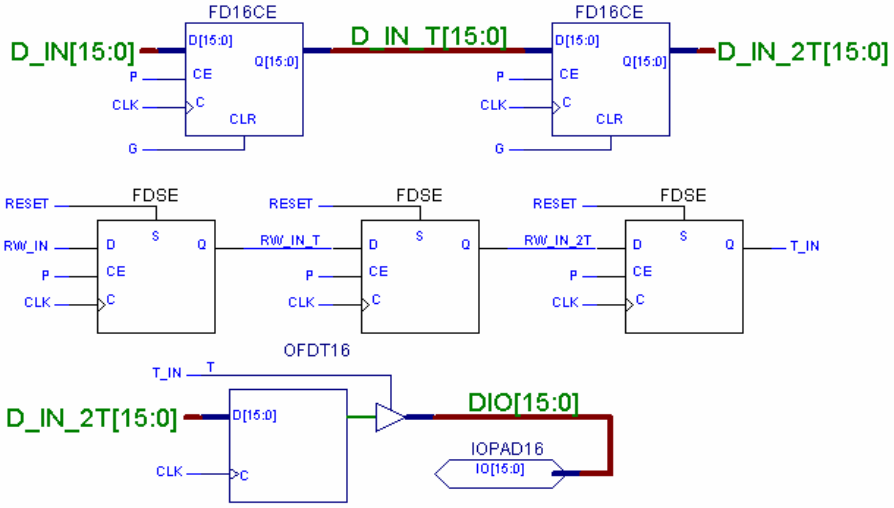


Рис. 8. Переключение двунаправленной шины данных
для синхронной памяти

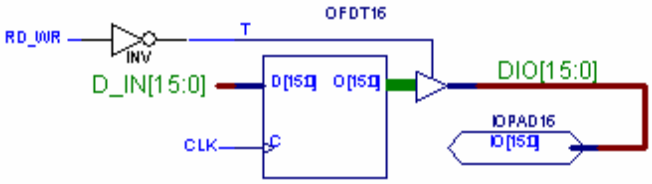


Рис. 9. Переключение двунаправленной шины данных

для асинхронной памяти

Схемы приема данных из памяти при чтении имеет следующий вид (рис.10 и 11):

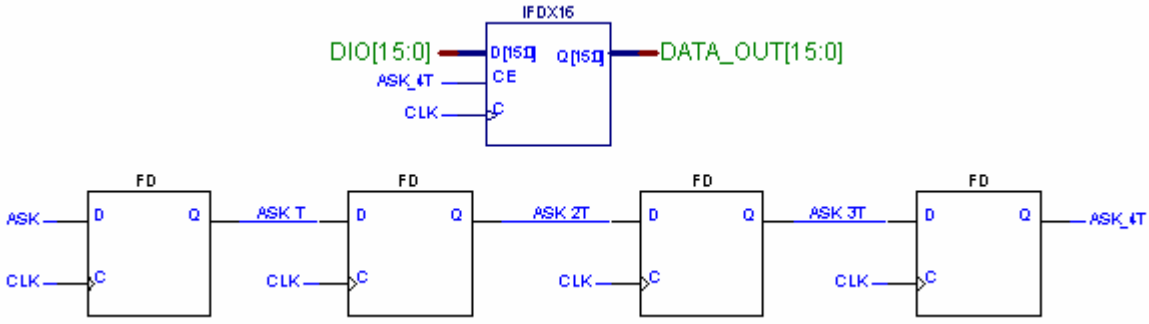


Рис. 10. Схема приема данных из синхронной памяти при чтении

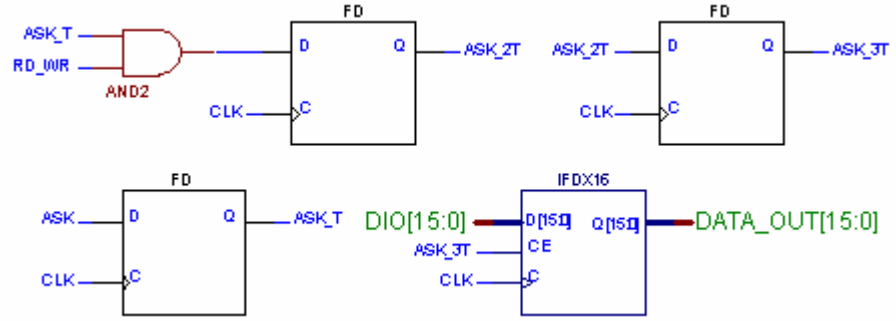


Рис.11. Схема приема данных из асинхронной памяти при чтении

Результаты использования ресурсов кристалла для одного из вариантов задания приняли вид:

Design Summary

Number of errors: 0
Number of warnings: 19
Number of Slices: 7 out of 1,200 1%
Number of Slices containing
unrelated logic: 0 out of 7 0 %
Number of Slice Flip Flops: 8 out of 2,400 1 %
Number of 4 input LUTs: 5 out of 2,400 1 %
Number of bonded IOBs: 69 out of 166 41 %
IOB Flip Flops: 50
Number of GCLKIOBs: 1 out of 4 25 %
Total equivalent gate count for design: 542
Additional JTAG gate count for IOBs: 3,360

Посмотреть отчет можно в файле *map.mrp* или в окне *Reports* менеджера проектов.

3. Порядок выполнения работы

3.1. Получить у преподавателя задание согласно номеру бригады (см. таблицу).

3.2. Построить блок приоритетов (см. примеры на рис. 1 и 2).

3.3. Выбрать тактовую частоту (для асинхронной памяти).
3.4. Построить схему установки адреса обращения к памяти.
3.5. Построить схему выдачи сигналов разрешения записи и чтения на память.
3.6. Построить схему переключения двунаправленной шины данных.
3.7. Построить схему приема данных из памяти.
3.8. Провести трассировку схемы.

Для этого необходимо в окне *Project Manager* запустить *Implementation*. При запуске разводки необходимо будет задать следующие опции:
Options→*Simulation*=OFF,
Options→*Implememntation*→*Optimize and Map*→*Trim Unconnected Logic*=OFF.

3.9. С помощью Tools→Implementation→Floorplanner запустить утилиту Floorplanner и посмотреть размещение проекта в кристалл.
3.8. Составить отчет.

Варианты заданий к лабораторной работе № 5

	Типа памяти	Тип приоритетов	Задержка	Количество каналов
1	Синхронная	Абсолютный	2 такта	3
2	Синхронная	Абсолютный	2 такта	4
3	Синхронная	Абсолютный	3 такта	3
4	Синхронная	Абсолютный	3 такта	4
5	Синхронная	Относительный	2 такта	3
6	Синхронная	Относительный	2 такта	4
7	Синхронная	Относительный	3 такта	3
8	Синхронная	Относительный	3 такта	4
9	Асинхронная	Абсолютный	20 ns	3
10	Асинхронная	Абсолютный	20 ns	4
11	Асинхронная	Абсолютный	30 ns	3
12	Асинхронная	Абсолютный	30 ns	4
13	Асинхронная	Относительный	20 ns	3
14	Асинхронная	Относительный	20 ns	4
15	Асинхронная	Относительный	30 ns	3

Приложение

ОПИСАНИЕ ИНТЕРФЕЙСА СХЕМОТЕХНИЧЕСКОГО РЕДАКТОРА

1.1. Определение схемы

Редактор обычно используется для создания изображения электрической схемы, которое содержит используемые компоненты и их соединения.

Схемотехническое изображение может быть использовано в твердой копии для документации или сборки схемы.

1.2. Физические и логические соединения

Физическое соединение - это графическая линия (называемая проводом), проводимая на схеме между двумя или более выводами.

Логическое соединение делается между двумя или более выводами без изображения провода. Это делается присвоением идентичных имен двум или более проводникам подсоединенными к выводам устройства.

1.3. Интерфейс схемотехнического редактора

Операции редактора управляются активацией иконок или кнопок на вертикальных или функциональных горизонтальных панелях.

1.3.1. Управление схемами проектов

Иконки, предназначенные для манипуляций со схемами проектов (*Project Management Icons*):

- Новая схема (*New Schematic*) - запускает новую схему.
- Открыть схему (*Open a Schematic*) - открывает одну из существующих схем.
- Сохранить (*Save*) - позволяет сохранить текущие схемные проекты.

Иконка активна только в случае, если в схеме были изменения после того, как она последний раз сохранена. В противном случае она останется серой или неактивной.

- Проекты (*Projects*) - отображает экран *Project Manager*.
- Печатать схему (*Print a Schematic*) - запускает типовую операцию печати *Windows*.

1.3.2. Операции исключения и наклейки

Иконки операций исключения, копирования и наклейки требуют установки редактора в режим *Select* нажатием иконки *Select and Drug*.

- Вырезать (*Cut*) - позволяет «забрать» выбранный элемент схемы в «карман». Сначала активизируйте режим *Select* нажатием на иконку *Select and Drug*. Далее переместите прямоугольник к выбранным элементам или выберите их нажатием клавиши *Shift* на каждом элементе. Нажатием на *Cut* удаляет выделенные элементы из схемы и помещает их в «карман». Все

провода, полностью остающиеся в удаляемой области, также копируются в «карман».

- Копировать (*Copy*) - позволяет копировать фрагмент схемы в «карман». Выберите копируемые элементы и нажмите на *Copy*.
- Клеить (*Paste*) - помещает содержимое «кармана» в выбранную позицию экрана. Первая активация *Paste* отображает отмеченное рамкой содержимое «кармана». Второе нажатие *Paste* помещает это содержимое в выбранную позицию.

1.3.3. Операции масштабирования

Группа иконок *Zoom Operations* изменяет масштаб выводимых на экран схем.

- Две кнопки масштаб (*Zoom*). Значение плюс (+) представляет опцию «Увеличить изображение» (*Zoom in*), а значок минус (-) - опцию «Уменьшить изображение» (*Zoom out*).
- Область масштабирования (*Zoom Area*) обеспечивает выбор фрагмента схемы, который будет отображен на полном экране. Первое нажатие *Zoom Area* активизирует курсор выбора. Поместите курсор в противоположный угол фрагмента. По освобождении мыши выбранная область расширится на весь экран.
- Масштаб полной страницы (*Full Page Zoom*) используется для отображения на экране полного рисунка схемы.

1.3.4. Операции редактирования

Иконки управления операциями редактирования:

- Отменить (*Undo*)- отменяет последнее схемное изменение и восстанавливает состояние схемы до него.
- Назначения (*Properties*) -позволяет изменять атрибуты проводов и обозначение элементов.
- Присоединить элемент (*Connect Symbol*) - соединяет выводы элементов с перекрытием с обозначениями конца провода (круг или квадрат с крестиком). Эти операции используются для замены деталей или присоединения проводов.
- Отсоединить элемент (*Disconnect Symbol*) - отсоединяет все провода от выбранного элемента и размещает обозначения концов отсоединенных проводов.
- Конец (*End*) - используется для разрыва проводов и шин в текущей позиции экрана. Провода могут разрываться введением обозначения их концов.
- Отмена (*Cancel*) - прерывает вычеркивание проводов и шин.

1.3.5. Операции ввода схемы

Иконки операций ввода схемы размещаются на левой стороне экрана. Однако они могут быть перемещены в другую позицию экрана.

- Элементная панель инструментов (*Symbol Toolbox*) - активизирует средства с обозначением устройства. Панель инструментов (*toolbox*) включает также иконки для выбора желаемых типов обозначений и позволяет поиск для выбранной детали или типов деталей.
- Рисовать провода (*Draw Wires*) - активизирует режим вычерчивания проводов. Нажатие на контакт начинает провод. Нажатие на другой контакт автоматически рисует провод. Двойное нажатие на пустом месте начинает провод с обозначением конца провода.
- Рисовать шины (*Draw Buses*) - позволяет рисовать шины нажатием в любой позиции экрана. Для окончания шины нажмите правую клавишу мыши или активируйте иконку *Ок*.
- Рисовать шинные контакты (*Draw Bus Tops*) - запускает вычерчивание соединений между шинами и контактами - ответвление шин. Нажатие на шину, а затем на обозначения контактов автоматически соединит эти контакты с шинами и ответвлениями.
- Добавить имя соединения или провода (*Add Net or Bus Name*) - используется для наименования проводов и шин. Нажатие на провод или шину в режиме *Select* окрасит выбранный элемент в красный цвет. Активизация иконки *Add Net or Bus Name* отобразит окно для наименования выбранных элементов.
- Элемент питания (*Power Symbol*) - позволяет прямое размещение обозначений питания на схеме. Иконка активна только, если также активна иконка *Draw Wires.*
- Терминал ввода/вывода (*I/O Terminal*) - позволяет наименование клемм ввода/вывода и задание их направления. Предварительно активизируется *Draw Wires.*
- Панель инструментов для графики (*Graphics Toolbox*) - позволяет вычерчивать графические элементы. Все элементы, рисуемые в этом режиме, не имеют электрических свойств. Этот режим используется для неэлектрических рисунков, для добавления информации, которую не нужно сохранять в электрической базе данных.

1.3.6. Общие операции

Выбор и перемещение икон - наиболее частая операция для выбора объектов на схеме и их обработки. Опция *Select* установлена автоматически для нескольких операций редактора.

Координаты. Строка статуса в нижнем левом углу отображает координаты курсора при его движении по схеме. Значение координат выдается в дюймах, миллиметрах или ссылках на зону. Выбор одного из этих типов - опция *Page Setup* в меню *File.*

Масштаб (*Ruler*). Отображение масштаба управляется опцией *View: Ruler*. Отображается слева (вертикальный) и наверху (горизонтальный) в каждом схемном окне.

Прямоугольники скроллинга размещены в правом нижнем углу каждого схемного окна. Операция позволяет плавное перемещение (скроллинг) по схеме в горизонтальном и вертикальном направлениях.

Оператор ЭДО ООО "Компания "Тензор"

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

СОГЛАСОВАНО **ФГБОУ ВО "РГРТУ", РГРТУ,** Литвинов Владимир Георгиевич, Заведующий кафедрой МНЭЛ **02.09.24** 14:18 (MSK) Простая подпись