

---

СИНТЕЗ И КОМПЬЮТЕРНЫЙ АНАЛИЗ  
ЭЛЕМЕНТОВ И УЗЛОВ ЦВМ НА FPGA С  
ИСПОЛЬЗОВАНИЕМ САПР QUARTUS

Методические указания  
по курсовому проектированию

# 1. ОСНОВНЫЕ ПОЛОЖЕНИЯ

## 1.1. Цель курсового проектирования

Целью курсового проектирования является освоение методов расчета, схемотехнического проектирования и конструирования элементов и блоков ЦВМ на базе программируемых логических интегральных схем (ПЛИС, далее FPGA).

При выполнении проекта автор должен разработать электронное устройство в соответствии с предложенной схемой и исходными данными, которое обеспечило бы заданную точность и качество работы. **Работоспособность устройства обязательно подтверждается моделированием с применением САПР Quartus.**

## 1.2. Задание на курсовое проектирование и содержание проекта

Задание на курсовое проектирование выдается индивидуально каждому студенту. В задании указывается тема и исходные данные для проектирования.

Разработка устройства включает в себя следующие этапы:

- составление и обоснование выбора функциональной схемы;
- разработку проектируемой схемы в САПР Quartus с использованием стандартных блоков, предоставляемых САПР (И, НЕ, дешифратор, триггер и т.д.) и описание ее работы с использованием временных диаграмм;
- разработку проектируемой схемы в САПР Quartus с использованием языка описания аппаратуры SystemVerilog и описание ее работы с использованием временных диаграмм;
- анализ двух подходов к разработке проектируемой схемы;
- обоснование и выбор FPGA на базе которого будет синтезироваться разрабатываемая схема;
- моделированием с применением САПР Quartus.
- назначение входов и выходов проектируемой схемы на выбранном FPGA.

### **1.3. Требования к отчету**

В результате выполнения курсового проекта к защите должны быть представлены:

- расчетно-пояснительная записка;
- реализованная схема с использованием стандартных блоков САПР Quartus;
- код на языке описания аппаратуры SystemVerilog;
- материалы моделирования (распечатанная программа, графики, таблицы и т.д.).

Пояснительная записка должна содержать результаты расчетов по всем этапам проектирования, обоснование принятых при разработке решений и необходимые для изложения графические материалы, в соответствии с разделом 1.2.

Записка должна быть оформлена в соответствии с требованиями ГОСТ 7.32–2001 на листах стандартного формата с текстом на одной стороне листа. Объем записки 15-30 страниц. Список используемых источников должен быть оформлен в соответствии с ГОСТ 7.32–2001, ГОСТ 7.1–2003.

### **1.4. Содержание пояснительной записки**

Содержание пояснительной записки должно быть разбито на разделы и подразделы, а при необходимости на пункты и подпункты.

В начале пояснительной записки помещается титульный лист, проектное задание и содержание. Последующее размещение материала в пояснительной записке рекомендуется провести в следующем порядке:

- список принятых буквенных обозначений;
- введение, в котором кратко формулируется поставленная задача, указывается назначение и область применения проектируемого устройства и анализируются исходные данные;

- выбор функциональной схемы, где сравниваются различные варианты построения устройства и выбирается лучший из них;
- составление схемы устройства с использованием стандартных блоков САПР Quartus и построение временных диаграмм в контрольных точках;
- составление схемы устройства с использованием языка описания аппаратуры SystemVerilog и построение временных диаграмм в контрольных точках;
- моделирование схемы устройства;
- обоснование и выбор FPGA;
- назначение входов и выходов проектируемой схемы на выбранном FPGA;
- заключение, в котором перечисляются основные результаты и дается оценка, насколько разработанное устройство удовлетворяет поставленным требованиям;
- список использованных источников;
- приложения.

В зависимости от особенностей разрабатываемой темы отдельные разделы допускается объединять, исключать либо вводить новые разделы.

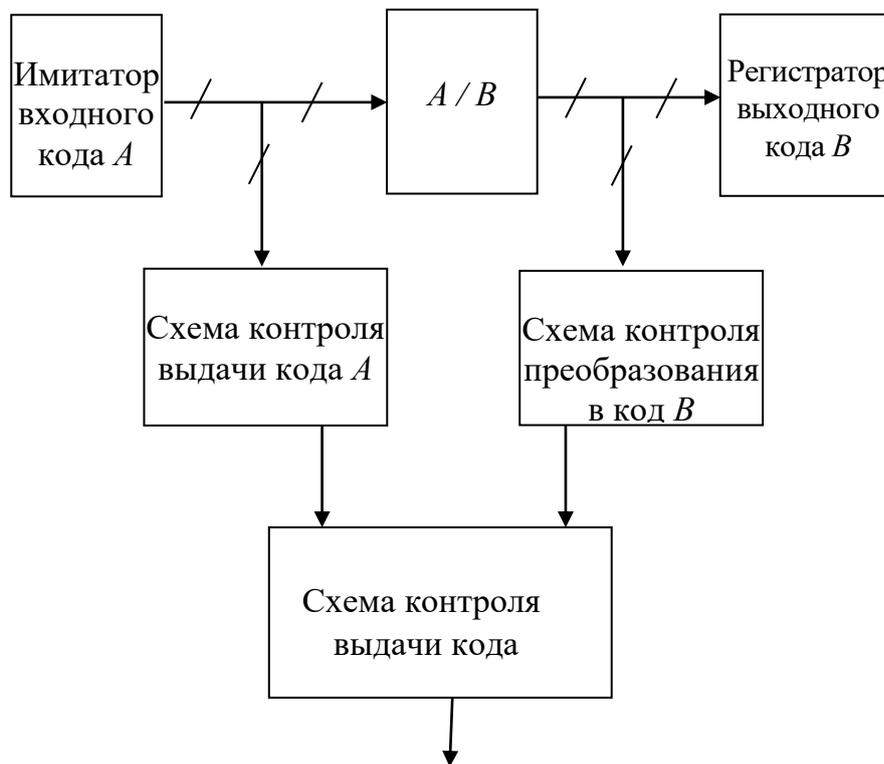
При выборе элементной базы для построения проектируемого устройства следует руководствоваться разделом 3 настоящих методических указаний.

Моделирование схемы устройства следует проводить с учетом рекомендаций раздела 3 настоящих методических указаний.

## 2. ВАРИАНТЫ ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ

Схема 1

Схема преобразователя кода



Таблицы кодов

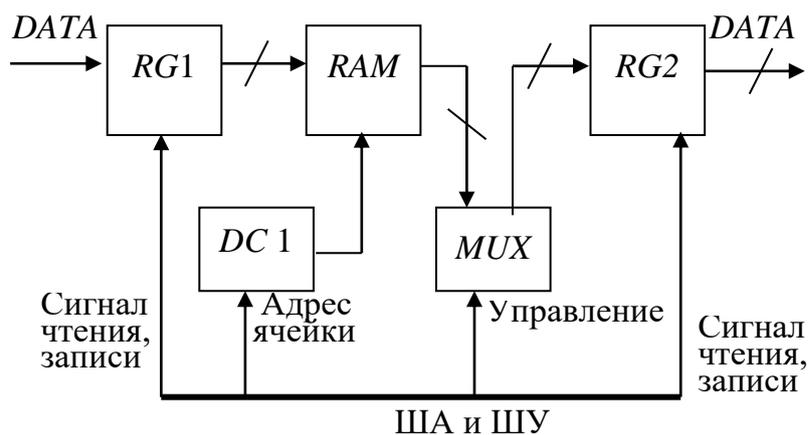
Десятичное число	Двоичный код	Двоично-десятичные коды						
		1	2	3	4	5	6	7
		8421	7421	5421	2421	5211	изб3	Грея
0	0000	0000	0000	0000	0000	0000	0011	0000
1	0001	0001	0001	0001	0001	0001	0100	0001
2	0010	0010	0010	0010	0010	0011	0101	0011
3	0011	0011	0011	0011	0011	0101	0110	0010
4	0100	0100	0100	0100	0100	0111	0111	0110
5	0101	0101	0101	1000	1011	1000	1000	0111
6	0110	0110	0110	1001	1100	1010	1001	0101
7	0111	0111	1000	1010	1101	1101	1010	0100
8	1000	1000	1001	1011	1110	1110	1011	1100
9	1001	1001	1010	1100	1111	1111	1100	1101

## Задания на проектирование

Номер варианта	Входной код КП	Выходной код КП	Номер варианта	Входной код КП	Выходной код КП
1	8421	5211	16	8421	Грея
2	7421	2421	17	8421	2421
3	2421	5211	18	7421	ИЗБ3
4	5421	ИЗБ3	19	5421	7421
5	ИЗБ3	Грея	20	Грея	ИЗБ3
6	5211	2421	21	2421	Грея
7	Грея	7421	22	5211	5421
8	7421	5211	23	ИЗБ3	8421
9	2421	ИЗБ3	24	Грея	5421
10	ИЗБ3	5421	25	ИЗБ3	7421
11	5421	2421	26	5421	Грея
12	5211	8421	27	5421	5211
13	7421	Грея	28	2421	7421
14	5211	7421	29	ИЗБ3	2421
15	Грея	2421	30	Грея	8421

### Схема 2

Схема записи и считывания из оперативной памяти

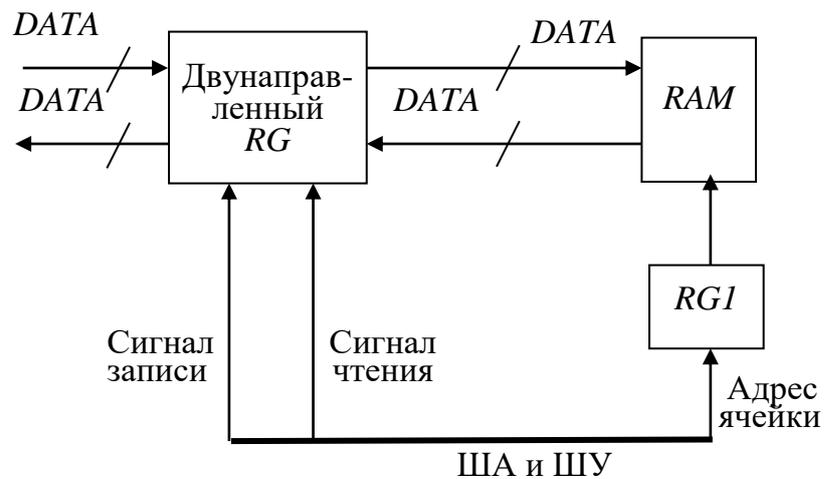


Вариант №	Разрядность	Адресность
1	4	128
2	8	64
3	16	32

По сигналам чтения и записи производится соответственно либо чтение, либо запись регистров  $RG1$  и  $RG2$ . Также из шины адреса поступает адрес ячейки, в которой производится запись или считывание данных.

### Схема 3

Блок для работы памяти с периферийными устройствами

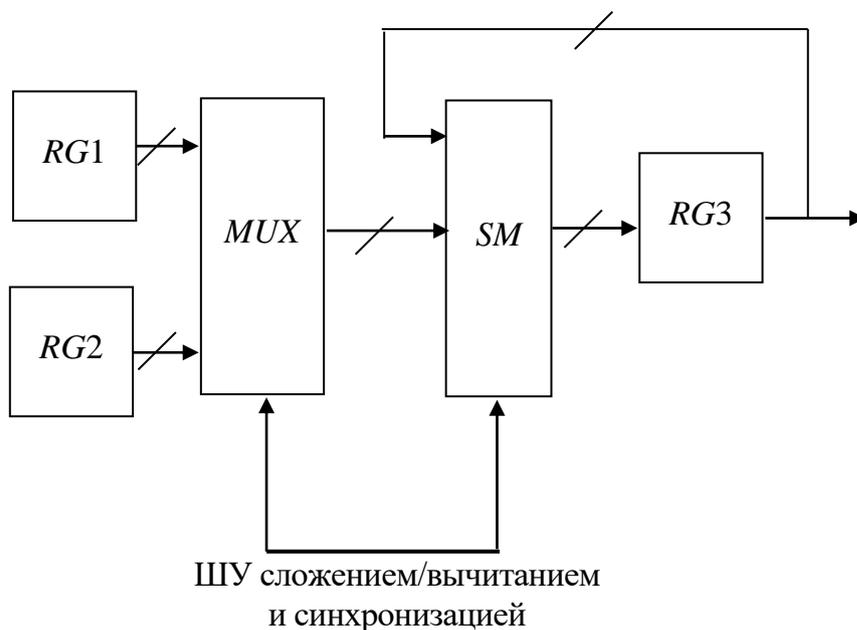


Вариант №	Разрядность	Адресность
1	4	128
2	8	64
3	16	32

По сигналам чтения/записи производится соответственно чтение/запись регистра  $RG$ . Также из шины адреса поступает адрес ячейки, в которой производится запись или считывание.

### Схема 4

#### Блок сложения и вычитания

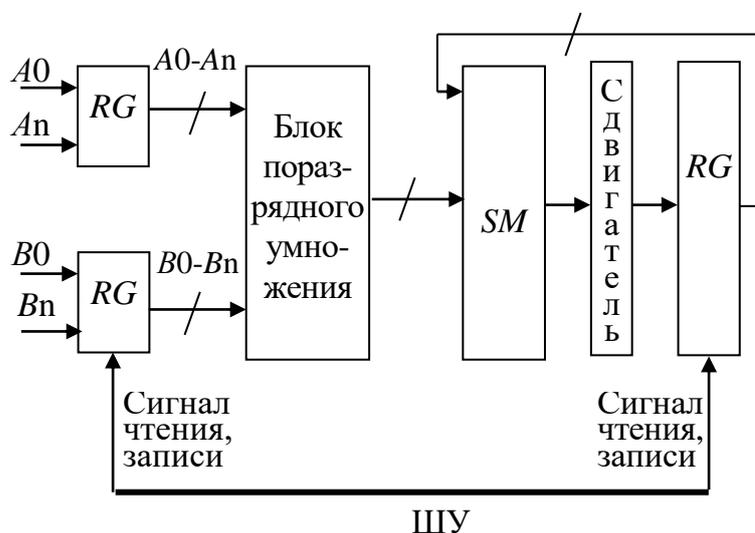


Вариант №	Разрядность
1	8
2	16

На вход блока подаются двоичные числа. Необходимо так выбрать разрядность сумматора и следующего за ним регистра, чтобы не было переполнения.

### Схема 5

#### Блок умножения

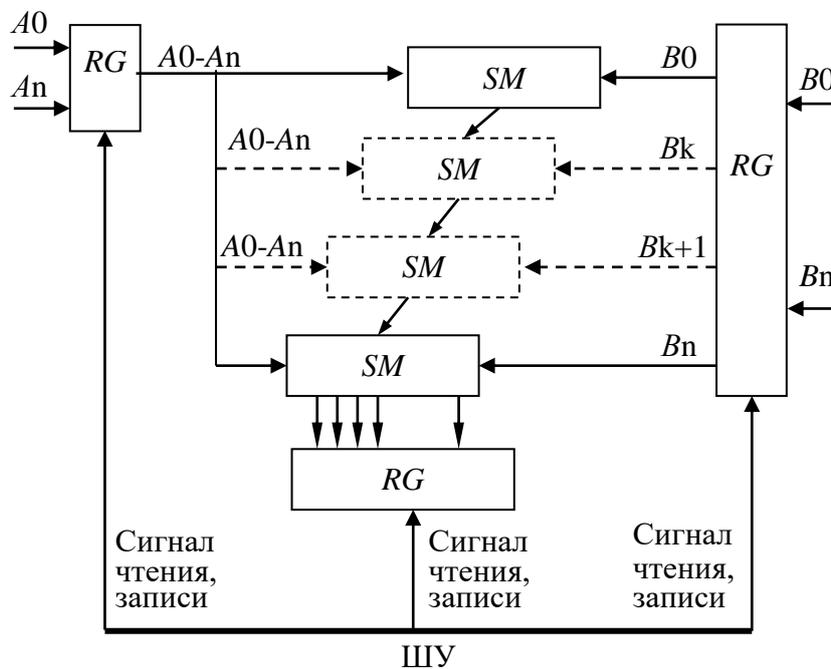


Вариант №	Разрядность
1	8
2	16

На вход блока подаются двоичные числа. Необходимо так выбрать разрядность сумматора и следующего за ним регистра, чтобы не было переполнения.

### Схема 6

#### Комбинационный блок умножения

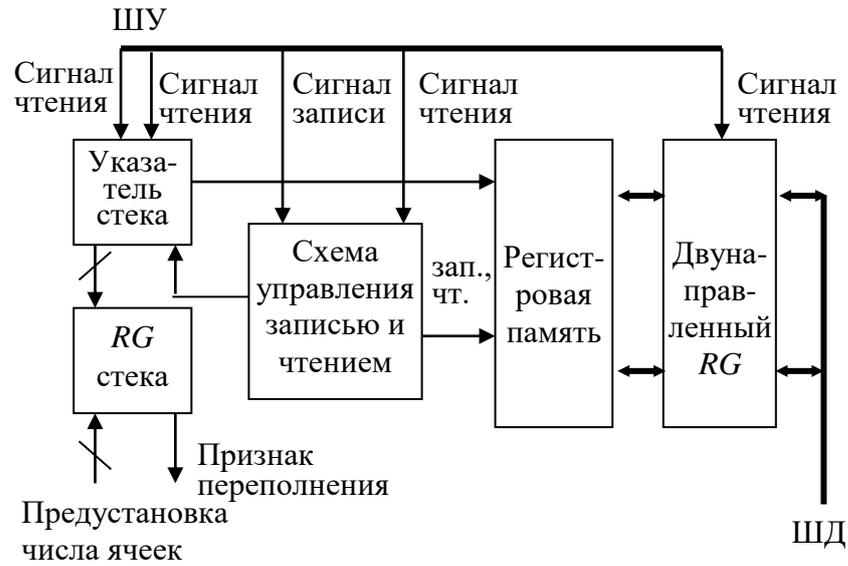


Вариант №	Разрядность
1	8
2	16

На вход блока подаются двоичные числа. Необходимо так выбрать разрядность сумматора и следующего за ним регистра, чтобы не было переполнения.

## Схема 7

### Стек LIFO

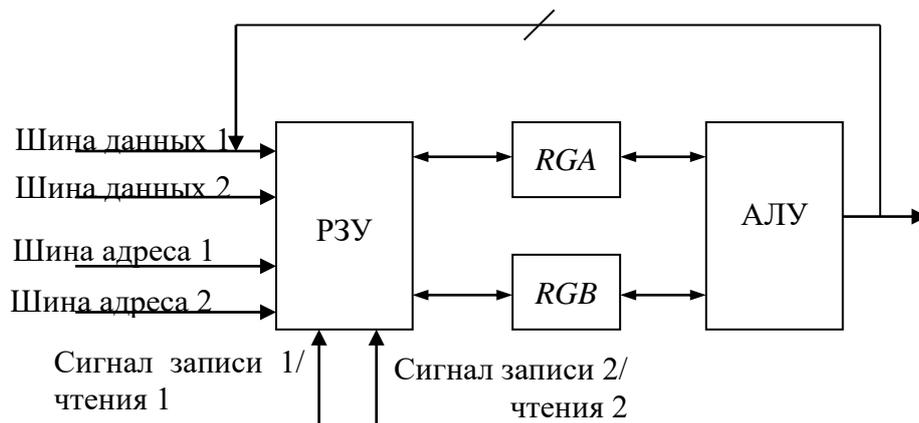


Вариант №	Разрядность	Адресность
1	8	128
2	16	64

Предустановка числа ячеек означает объем стека. Схема сравнения разрешает прохождение сигнала записи. Анализируется счетчик. Если счетчик +1, то схема управления заносит в стек. Если >1, то схема управления выдает сигнал на сдвиг вниз с задержкой сигнала записи. По сигналу чтения производится чтение, и после чтения схемой управления вырабатывается сигнал сдвига вверх. Схема сравнения необходима для анализа состояния заполнения стека.

## Схема 8

### Тактируемое регистровое запоминающее устройство



### Вариант 1

Тактируемое регистровое запоминающее устройство на 4 слова с одним входом, одним выходом и разрешением записи.

### Вариант 2

Тактируемое регистровое запоминающее устройство на 4 слова с одним входом, двумя выходами и разрешением записи.

### Вариант 3

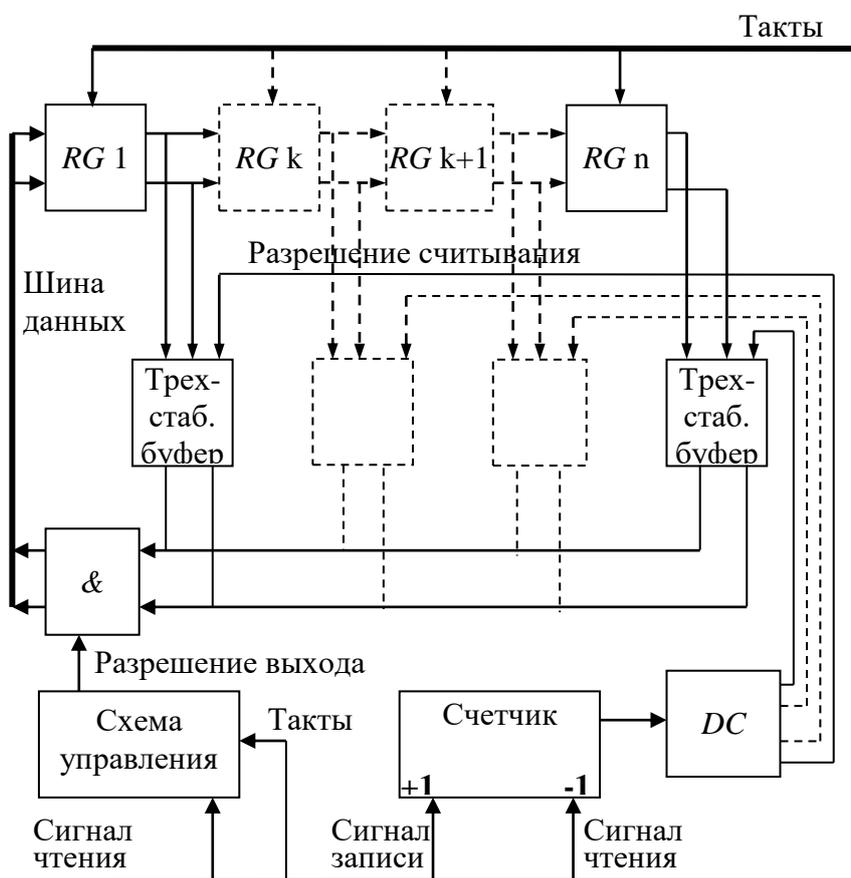
Тактируемое регистровое запоминающее устройство на 4 слова с двумя входами, одним выходом и разрешением записи.

### Вариант 4

Тактируемое регистровое запоминающее устройство на 4 слова с двумя входами, двумя выходами и разрешением записи по каждому входу.

*Примечание:* В АЛУ по заданию преподавателя выполняется одна из операций: суммирование, дизъюнкция, конъюнкция, сложение по MOD2.

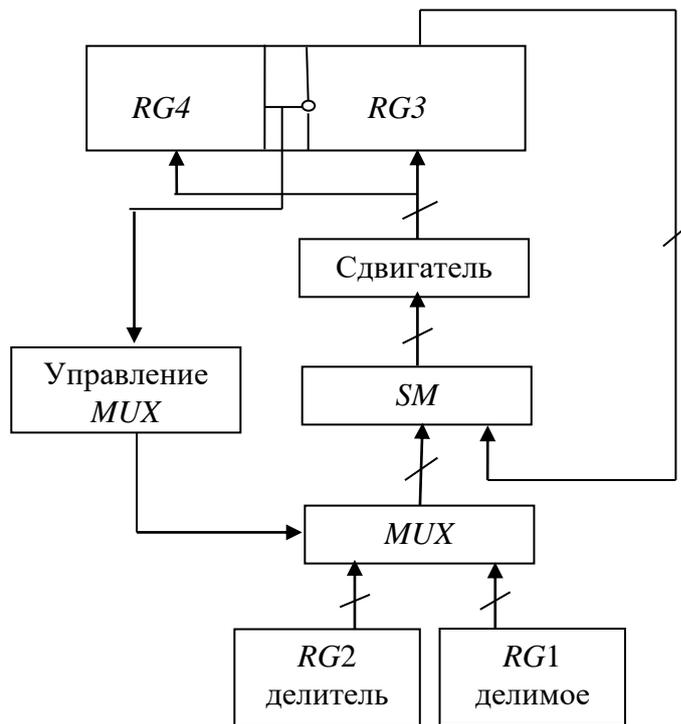
Схема 9  
Стек FIFO



Вариант №	Разрядность	Адресность
1	8	128
2	16	64
3	32	32

Схема 10

Блок деления



Вариант №	Разрядность
1	8
2	16

На вход блока подаются двоичные числа. Необходимо так выбрать разрядность АЛУ, чтобы не было переполнения и на количество.

### 3. РЕКОМЕНДАЦИИ ПО МОДЕЛИРОВАНИЮ И РАЗРАБОТКЕ

#### 3.1. Синтез схемы в САПР Quartus с использованием стандартных блоков

В качестве примера для первой части курсового проекта синтезируем вычитающий 4-х битный последовательный счетчик на T-триггерах в САПР Quartus (Quartus Prime 17.1) Lite Edition [1]. Для начала создадим новый проект (**File** ⇒ **New Project Wizard**), как это показано на рис. 1:

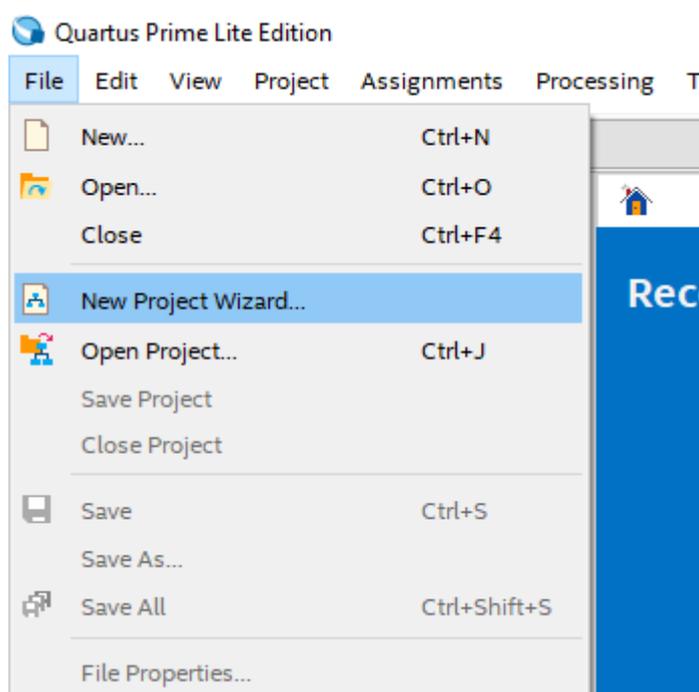


Рисунок 1 – Создание нового проекта

Следующим шагом выберем директорию создаваемого проекта и его название (см. рис. 2). На последующих шагах, вплоть до выбора FPGA под который САПР будет синтезировать разрабатываемый нами счетчик, жмем **Next**. Выберем любое FPGA семейства **Cyclone IV E**, как приведено на рисунке 3. На данном шаге можно завершить создание нового проекта, нажав на **Finish**.

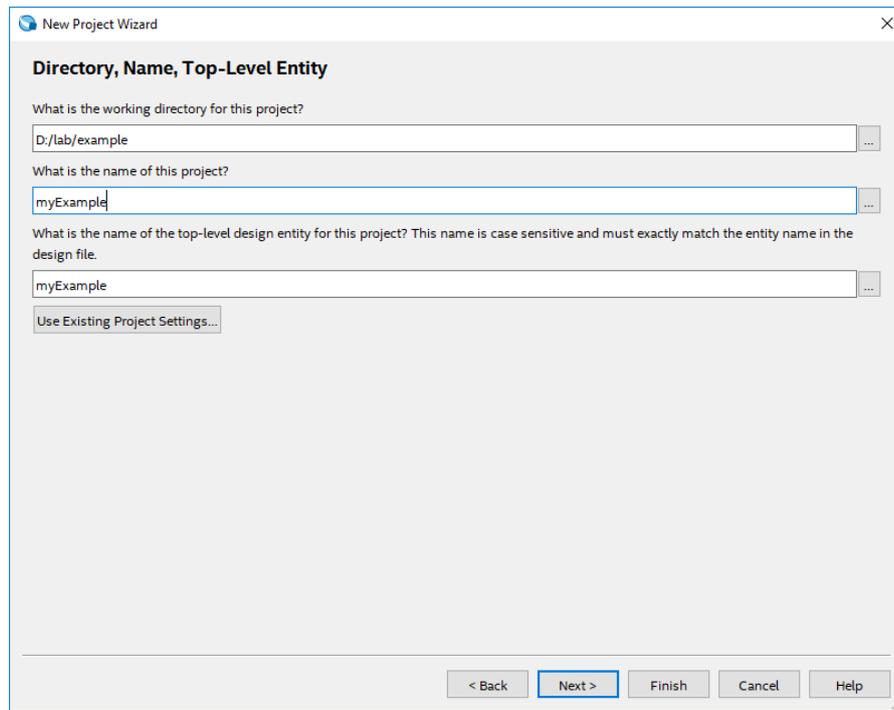


Рисунок 2 – Выбор директории и названия создаваемого проекта

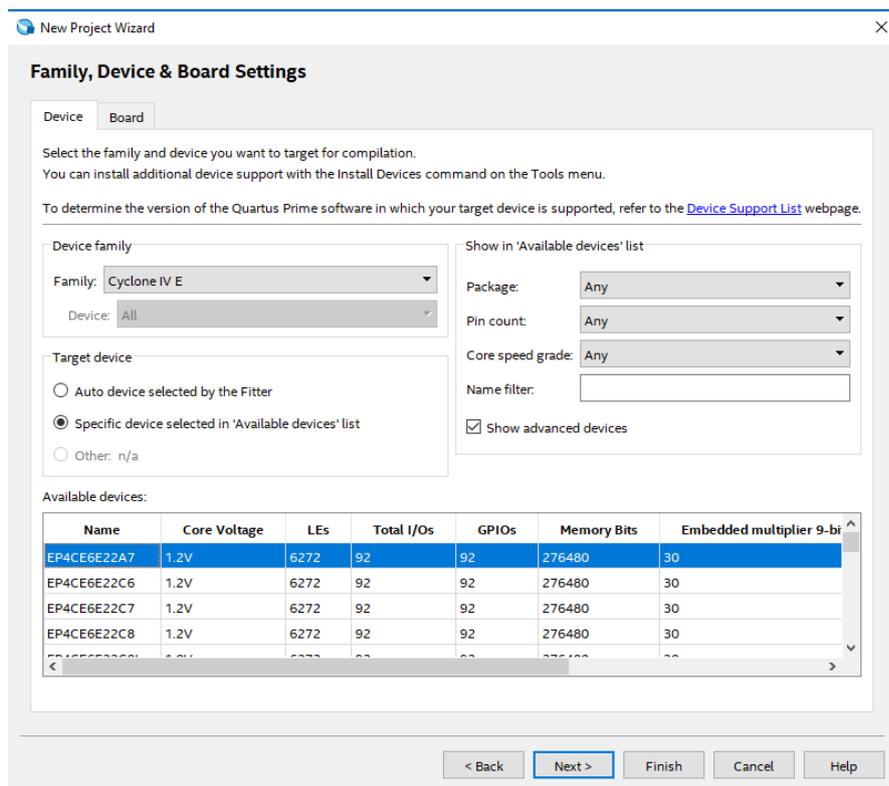


Рисунок 3 – Выбор FPGA

После создания проекта необходимо создать Block Diagram (**File** ⇒ **New** ⇒ **Block Diagram**, см. рис. 4), на котором будем производить разработку счетчика, посредством стандартных блоков САПР (Т-триггер).

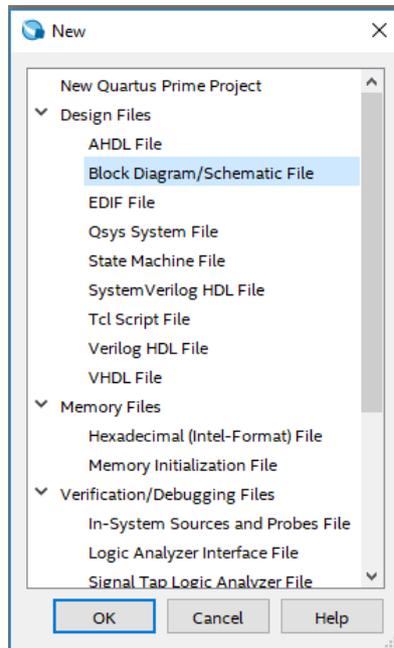


Рисунок 4 – Создание Block Diagram

Block Diagram на рабочей области примет следующий вид:

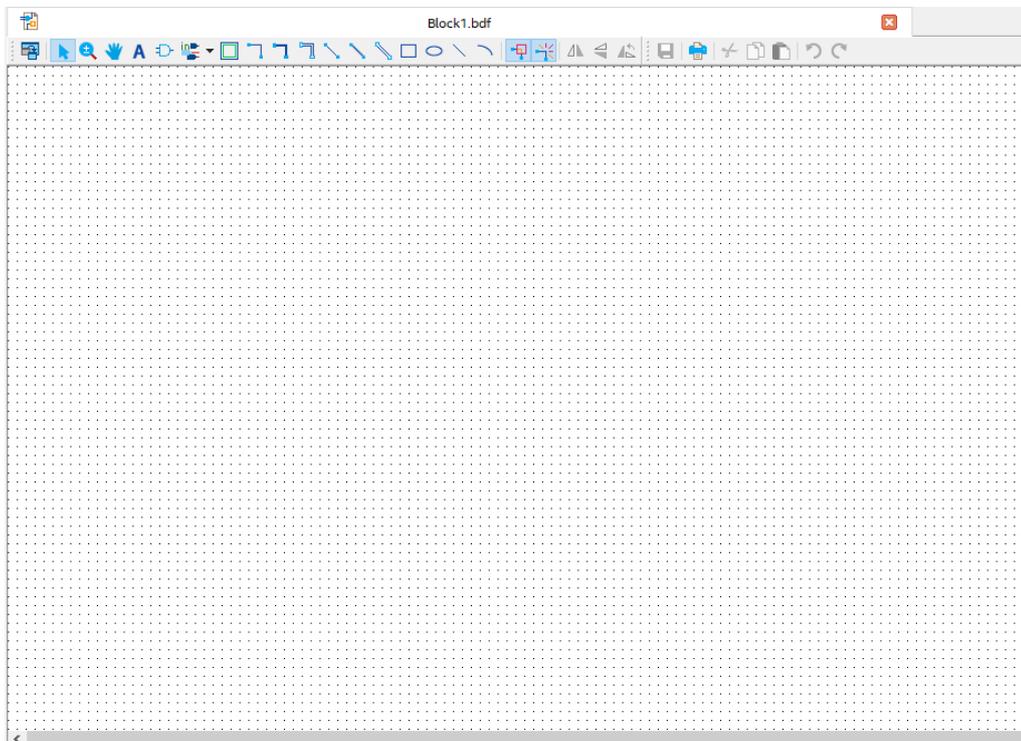


Рисунок 5 – Block Diagram

Добавим 4 синхронных Т-триггера на Block Diagram. Для этого необходимо вызвать **Symbol Tools** (два щелчка мышью на рабочей области Block Diagram или как показано на рис. 6):



Рисунок 6 – Вызов Symbol Tools

В появившейся области следует найти блок Т-триггера (рис. 7). В САПР Quartus Т-триггер обозначается как **tff** и расположен по следующему пути: **Primitives** ⇒ **Storage** ⇒ **tff**. После того, как Т-триггер выбран жмем ОК и помещаем его на Block Diagram.

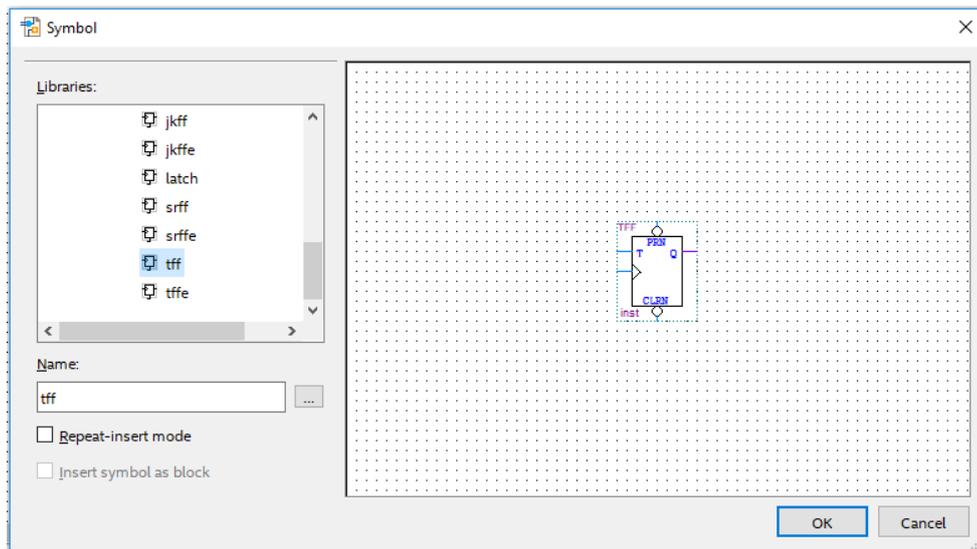


Рисунок 7 – Выбор Т-триггера в Symbol Tools

Скопируем размещенный на Block Diagram Т-триггер несколько раз и соединим их как показано на рис. 8:

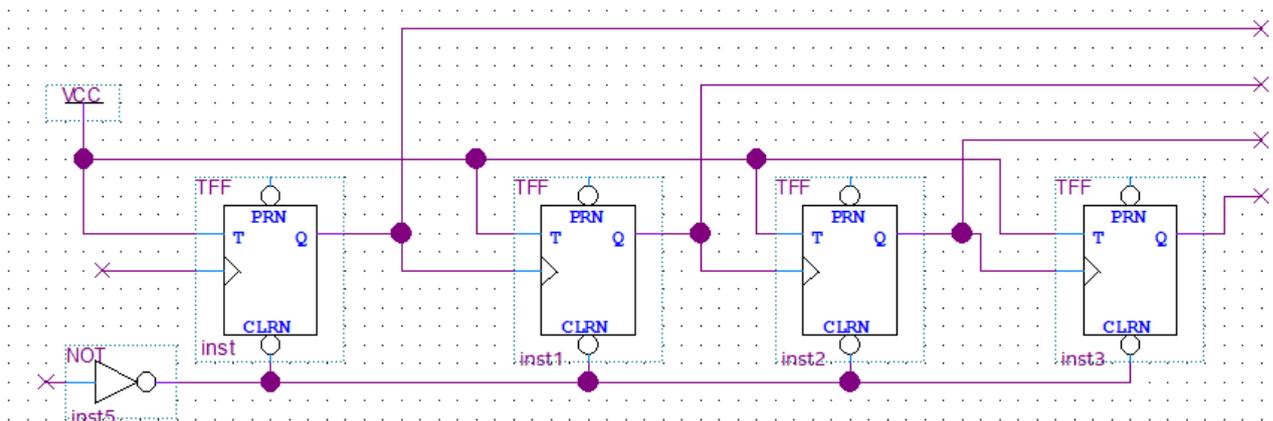


Рисунок 8 – Разработка 4-х битного вычитающего счетчика

Теперь необходимо добавить порты входов и выходов (I/O Ports) через Symbol Tools. Они расположены по следующему пути: **Primitives** ⇒ **Pin**. Законченный вариант разрабатываемого нами счетчика представлен на следующем рисунке:

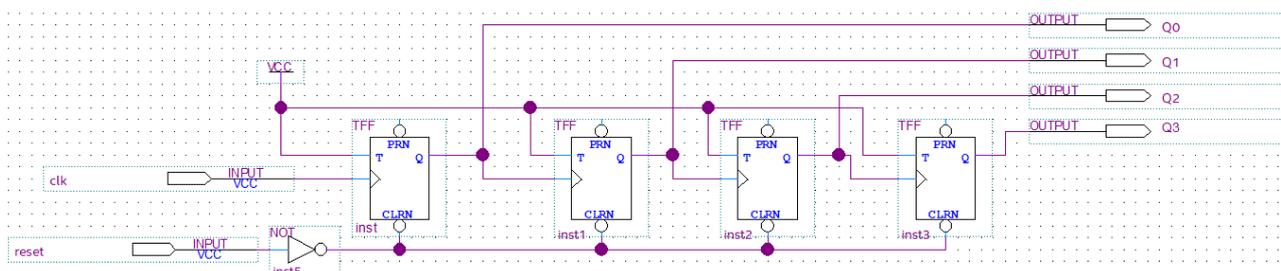


Рисунок 9 – Законченный вариант разрабатываемого счетчика

Сохраним его под названием **myCounter**. Следующим шагом скомпилируем наш проект нажав на . В случае возникновения ошибки «**Error (12007): Top-level design entity "myExample" is undefined**» необходимо установить **Top-Level** проекта как показано на рисунке 10:

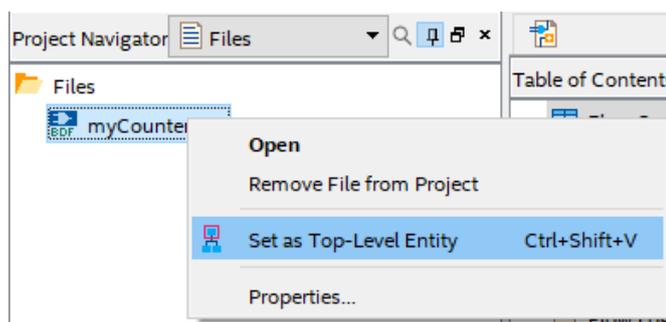


Рисунок 10 – Установка Top-level проекта

После успешной компиляции проекта необходимо произвести симуляцию разработанного счетчика. Для этого САПР Quartus предоставляет внутренний инструментарий. Чтобы им воспользоваться создадим файл временной диаграммы (далее Waveform) через **File** ⇒ **New** (рис. 11):

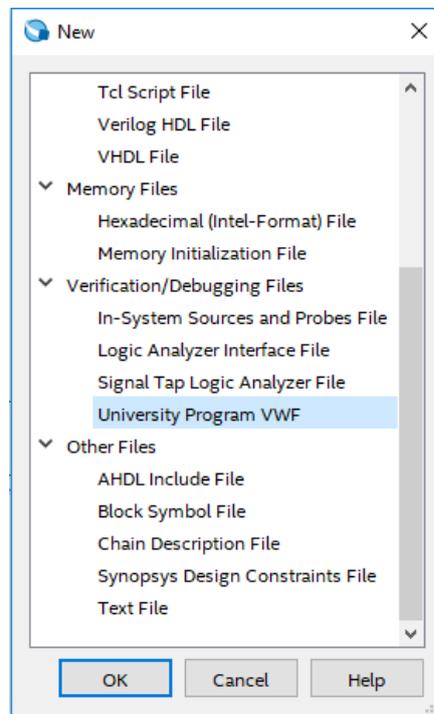


Рисунок 11 – Создание файла Waveform

Добавим порты входов и выходов из Block Diagram на Waveform, щелкнув правой кнопкой мыши и выбрав «**Insert Node or Bus**»:

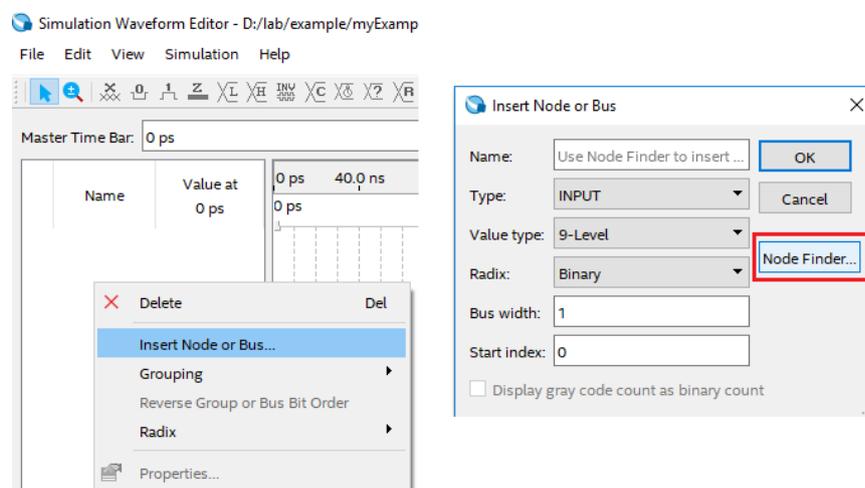


Рисунок 12 – Добавление сигналов на Waveform

Шаги последующих действий приведены на рисунке ниже:

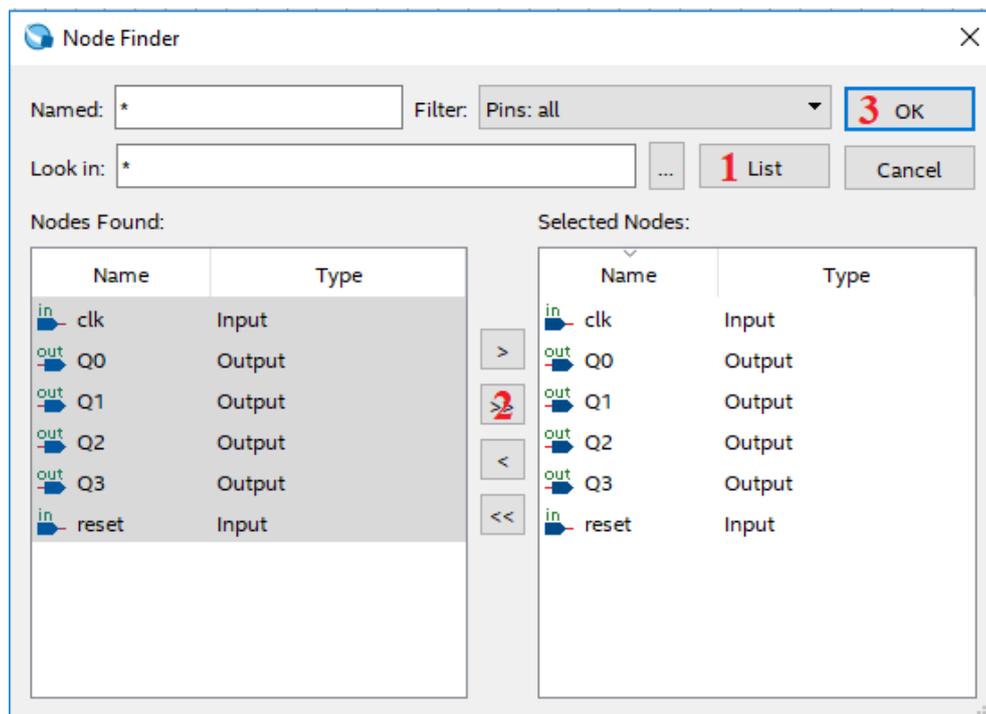


Рисунок 13 – Добавление сигналов на Waveform

После проделанных действий Waveform будет иметь вид представленный на рисунке 14:

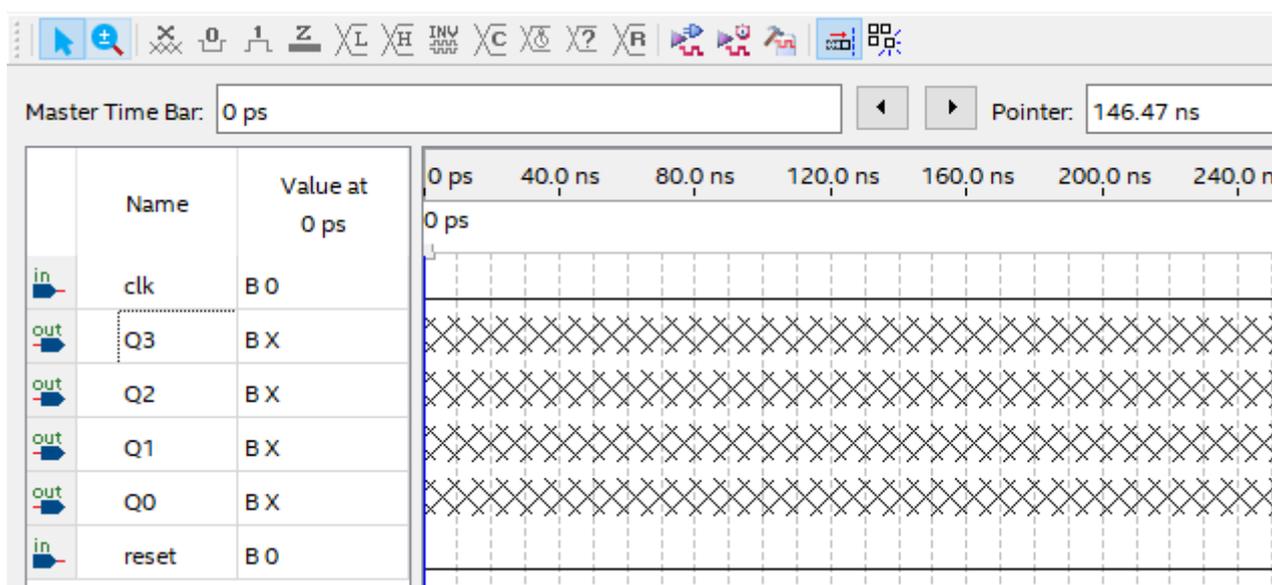


Рисунок 14 – Waveform вычитающего счетчика

Сгруппируем выходные сигналы вычитающего счетчика:

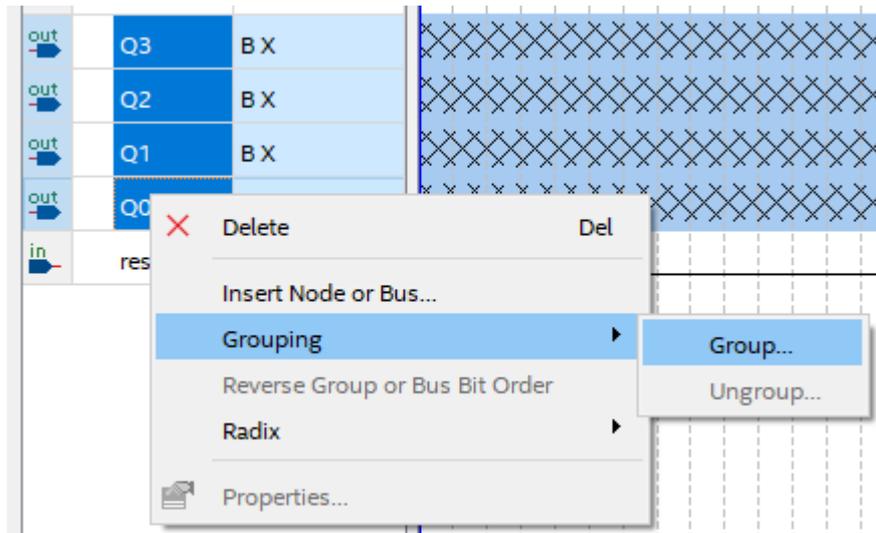


Рисунок 15 – Группировка выходных сигналов

После проделанных действий Waveform будет иметь следующий вид:

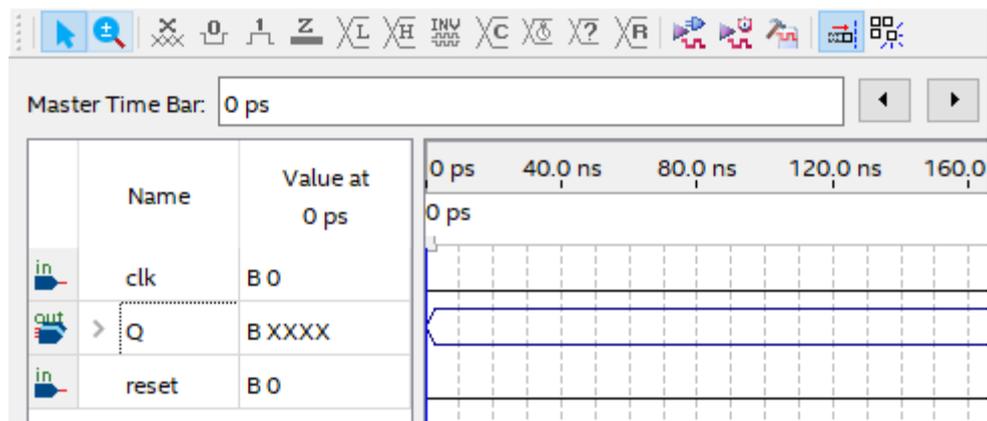


Рисунок 16 – Waveform вычитающего счетчика

Следующим шагом установим период (в нс) с которым будет осуществляться счет:

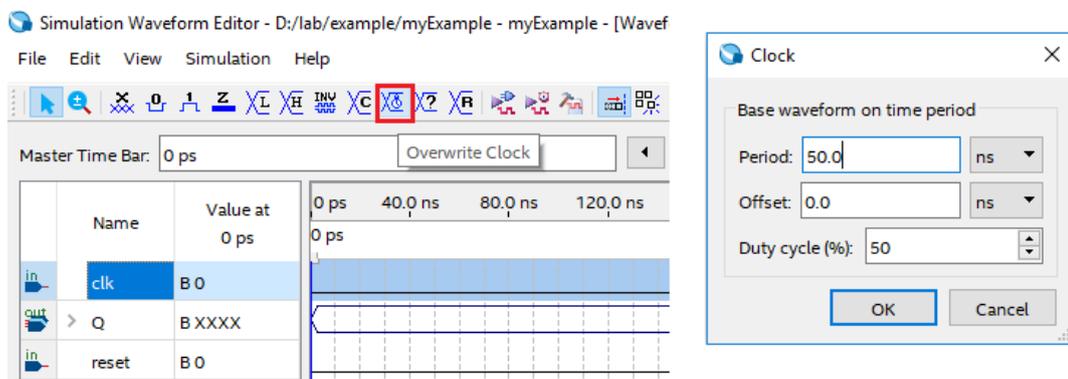


Рисунок 17 – Установка периодичности счета

В итоге Waveform должен иметь вид, представленный на рисунке ниже:

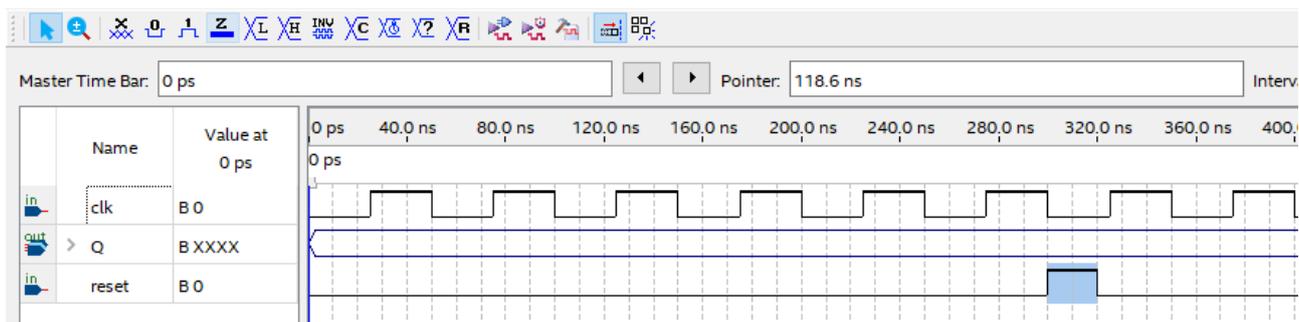


Рисунок 18 – Waveform вычитающего счетчика перед симуляцией

Запустим функциональную симуляцию, нажав на :

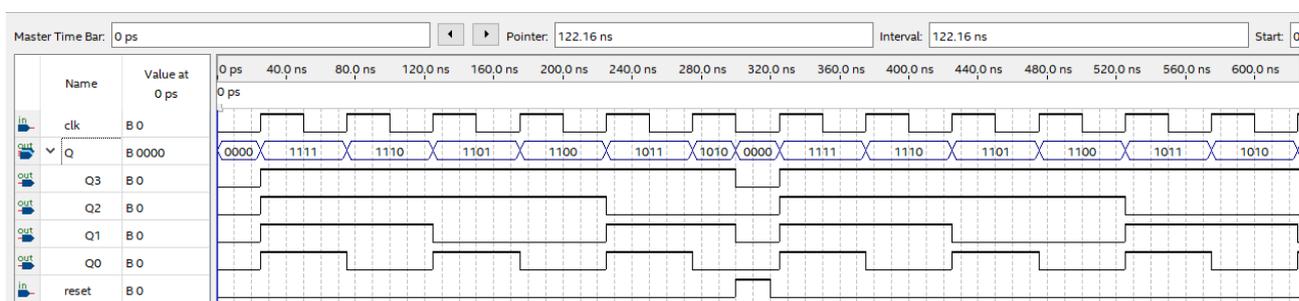


Рисунок 19 – Симуляция вычитающего счетчика

По умолчанию на выходной шине **Q** данные выводятся в двоичном представлении. Для удобства проверки правильности работы установим свойства шины **Q** таким образом, чтобы значения отображались в виде без знаковых целых чисел:

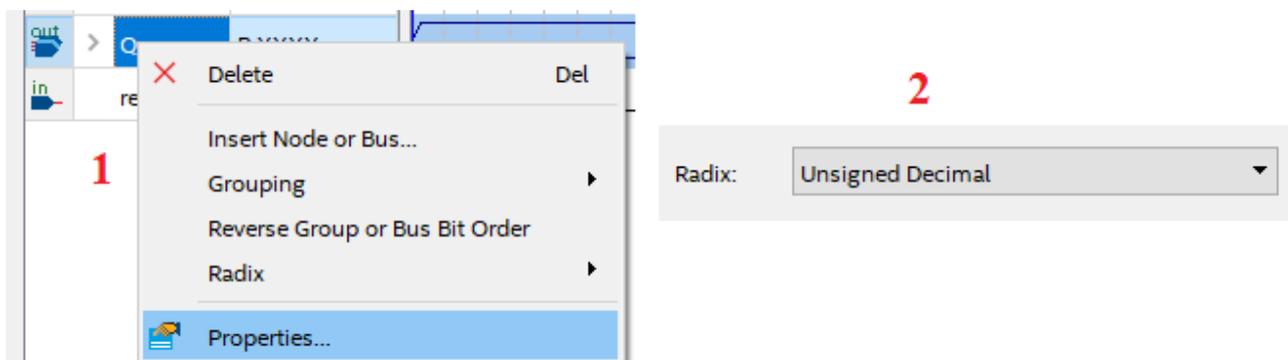


Рисунок 20 – Настройка свойств шины

Таким образом результирующая Waveform будет иметь следующий вид:

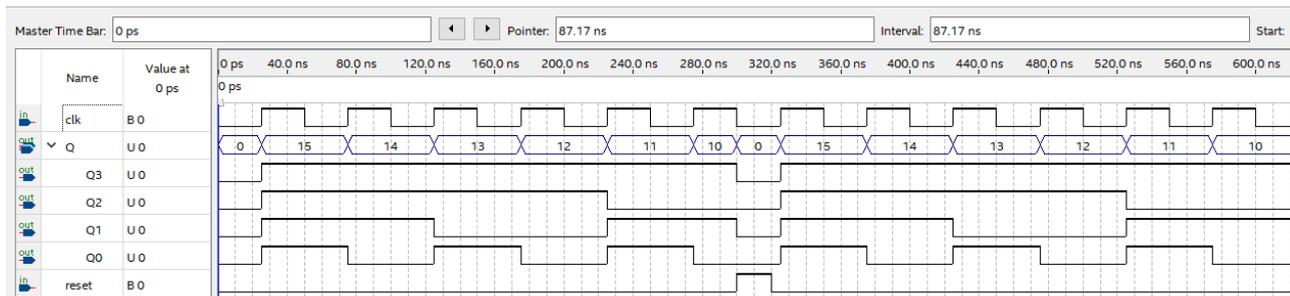


Рисунок 21 – Симуляция вычитающего счетчика

### 3.2. Пример создания собственных блоков для САПР Quartus

Рассмотрим пример создания собственного блока для Block Diagram на основе уже разработанного вычитающего счетчика. Удобство данного подхода заключается в том, что разрабатываемая схема, если состоит из множества частей, приобретает более компактный вид.

Выберем Block Diagram с разработанным вычитающим счетчиком и сделаем следующие действия: **File** ⇒ **Create/Update** ⇒ **Create Symbol Files for Current File** (см. рис. 22) и сохраним его с именем **CounterBlock**.

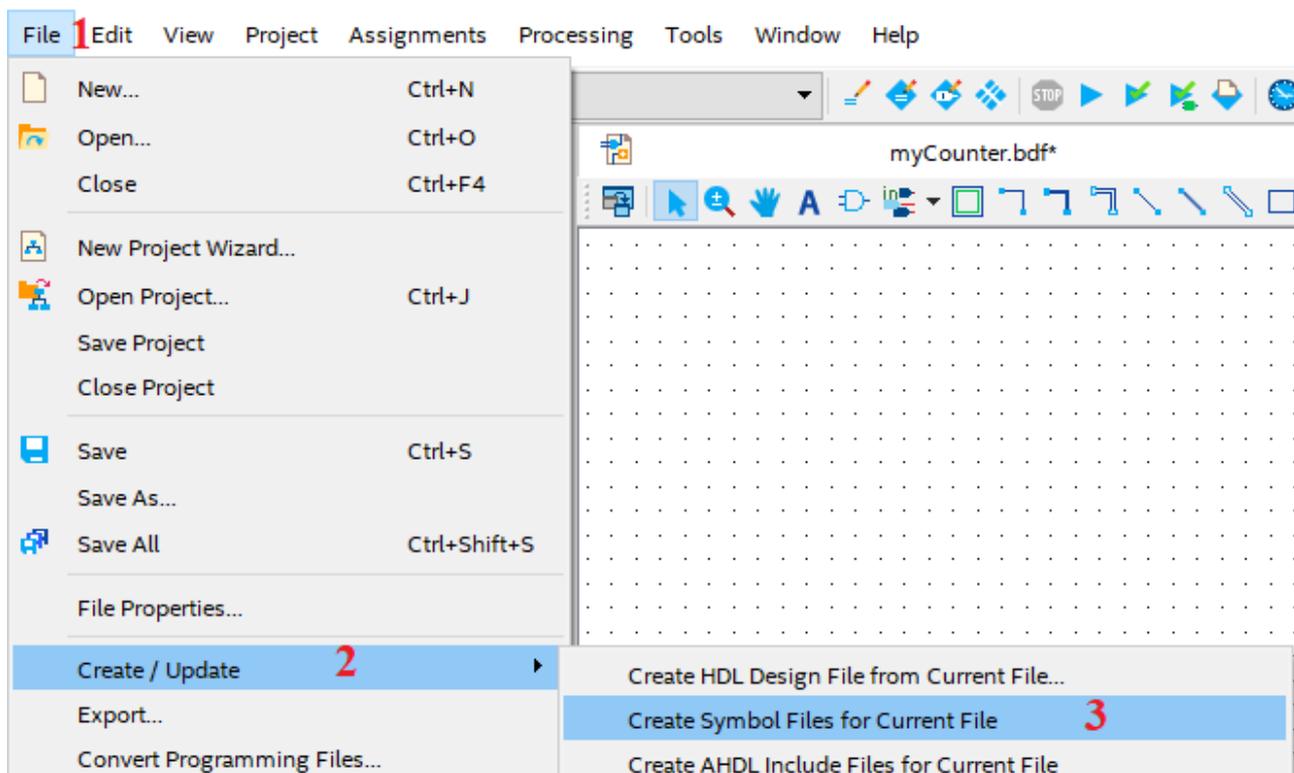


Рисунок 22 – Шаги создания собственного блока

Теперь при создании новой Block Diagram в Symbol Tools будет доступен сгенерированный САПР блок вычитающего счетчика:

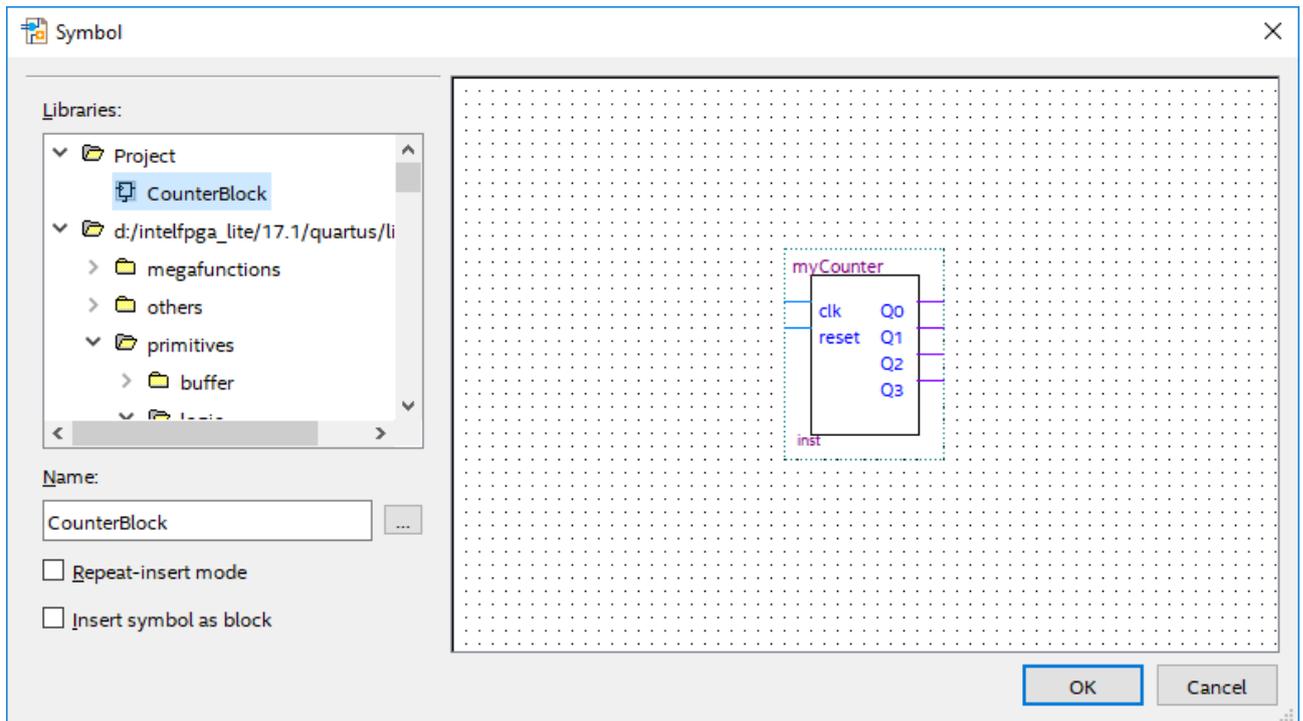


Рисунок 23 – Вид сгенерированного блока

Зачастую бывает удобнее объединить выходные сигналы с одним именем в шину. Для этого на Block Diagram добавим выходной порт и через свойства ему и выходным сигналам блока **myCounter** зададим необходимые значения, как показано на рисунке 24:

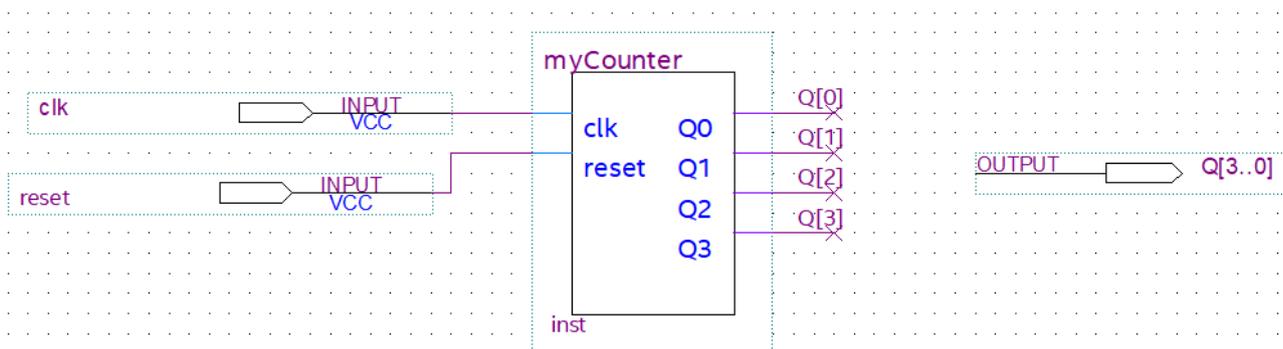


Рисунок 23 – Объединение выходных сигналов блока myCounter в шину Q

### 3.3. Синтез схемы в САПР Quartus с использованием языка описания аппаратуры SystemVerilog

Во второй части курсового проекта (КП) необходимо реализовать ту же самую схему, что и в первой части, но уже с использованием языка описания аппаратуры SystemVerilog. В ходе написания первой части КП могут быть проблемы, связанные с тем, что необходимый элемент среди стандартных блоков САПР не был найден или отсутствует. В этом случае не запрещается реализовать данный элемент посредством SystemVerilog.

Создадим новый проект (см. раздел 3.1) и реализуем вычитающий счетчик на SystemVerilog. Для этого необходимо проделать следующие шаги (см. рис. 24): **File ⇒ New ⇒ SystemVerilog HDL File.**

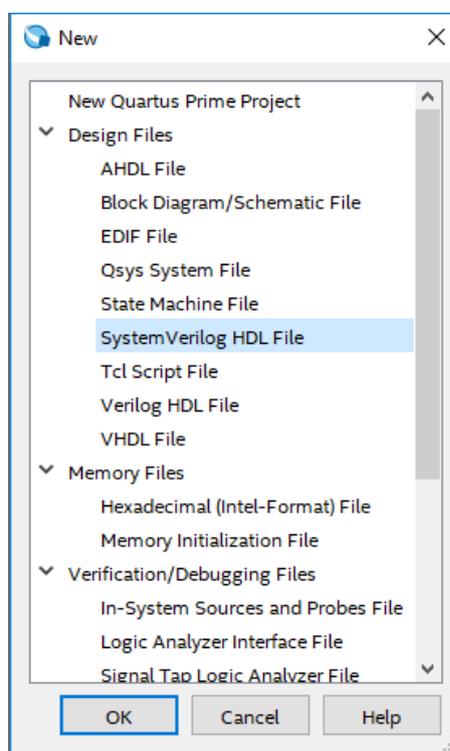


Рисунок 24 – Создание файла SystemVerilog

В открывшемся файле напишем следующий код:

```

module countersV(input logic clk, reset,
                 output logic[3:0] Q);

always @(posedge clk or posedge reset)
begin
if (reset)
    Q = 0;
else
    Q = Q - 1;
end
endmodule

```

Рисунок 25 – Листинг кода на SystemVerilog

Сохраним созданный файл с таким же именем, как и модуль описанный в нем (**counterSV**), после чего сгенерируем блок из написанного кода (см. раздел 3.2) для его размещения на Block Diagram. Таким образом завершённый проект вычитающего счетчика на SystemVerilog будет иметь вид:

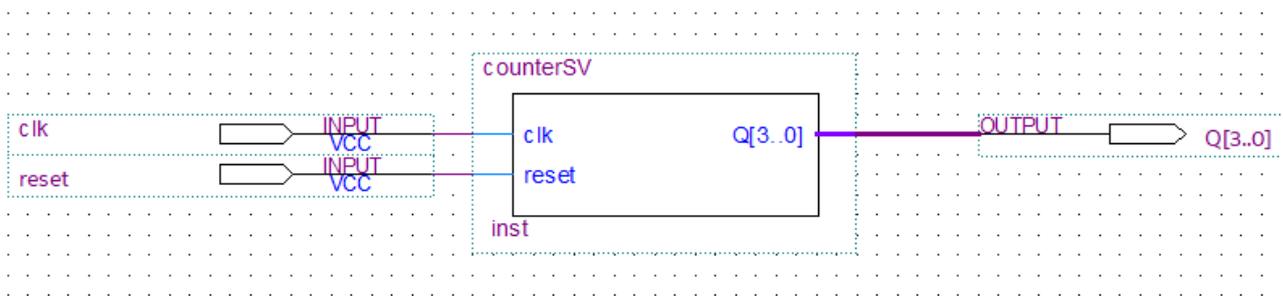


Рисунок 26 – Сгенерированный блок из кода на SystemVerilog

Временная диаграмма работы вычитающего счетчика на SystemVerilog представлена на рис. 27:

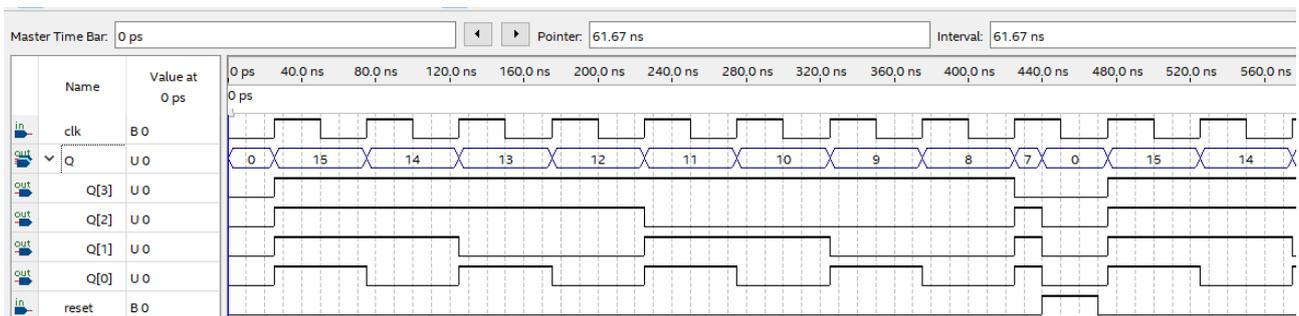


Рисунок 27 – Симуляция вычитающего счетчика на SystemVerilog

Посмотрим, как САПР на основе кода на SystemVerilog синтезировала вычитающий счетчик. Для этого воспользуемся утилитой, входящей в состав

САПР Quartus, RTL Viewer (см. рис. 28): **Tools** ⇒ **Netlist Viewers** ⇒ **RTL Viewer**.

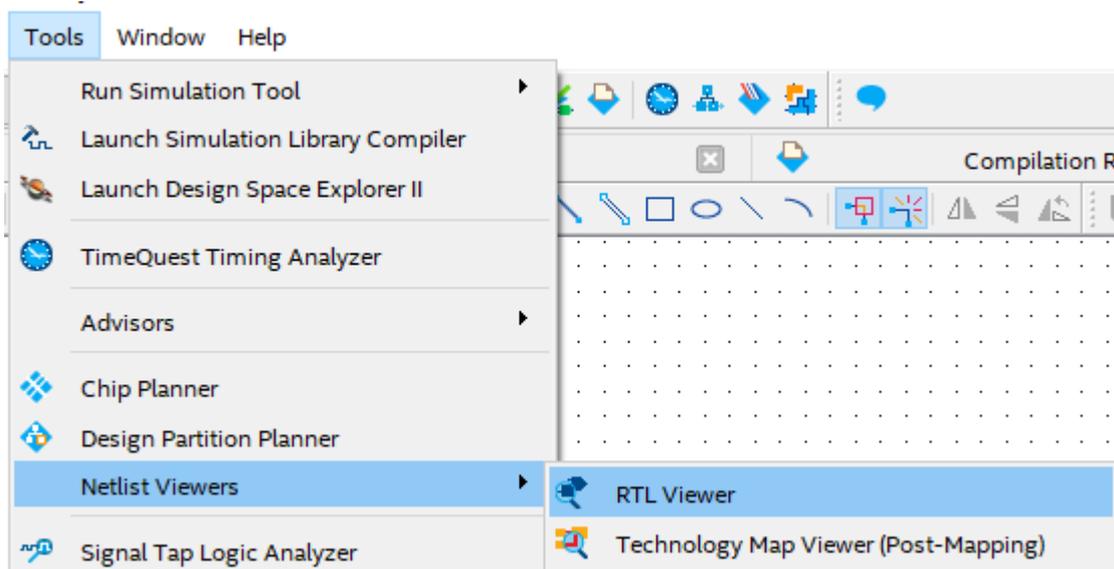


Рисунок 28 – Вызов RTL Viewer

Из получившейся развертки можно заметить, что САПР представило вычитающий счетчик как 5-ти битный сумматор и регистр из 4-х D-триггеров. Выход регистра подается на сумматор по шине **A**. При этом младший разряд шины **A** ( $A[0]$ ) всегда равен 1, а второй константный оператор, подаваемый по шине **B** равен 29 в десятичном представлении.

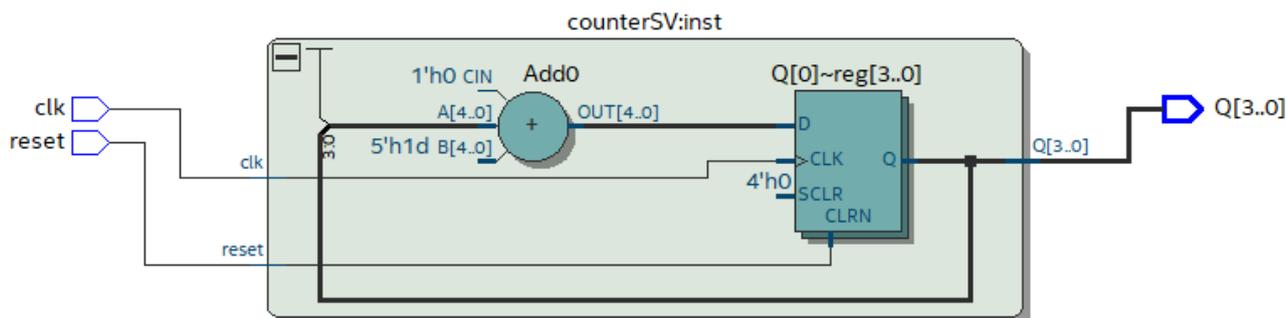


Рисунок 29 –RTL развертка проекта

### 3.4. Анализ затрачиваемых ресурсов и выбор FPGA

После компиляции проекта САПР Quartus выводит оценку затрачиваемых FPGA ресурсов. Как можно видеть из рисунка 30 в случае, когда вычитающий

счетчик описывался на SystemVerilog затрачивается меньше логических элементов при синтезе схемы.

Flow Summary		Flow Summary	
Flow Status	Successful - Tue May 01 16:33:54 2018	Flow Status	Successful - Tue May 01 16:31:21 2018
Quartus Prime Version	17.1.0 Build 590 10/25/2017 SJ Lite Edition	Quartus Prime Version	17.1.0 Build 590 10/25/2017 SJ Lite Edition
Revision Name	myExample	Revision Name	myCounter_SV
Top-level Entity Name	myExample	Top-level Entity Name	myCounter_SV
Family	Cyclone IV E	Family	Cyclone IV E
Device	EP4CE6E22A7	Device	EP4CE6E22A7
Timing Models	Final	Timing Models	Final
Total logic elements	7 / 6,272 (< 1 %)	Total logic elements	4 / 6,272 (< 1 %)
Total registers	4	Total registers	4
Total pins	6 / 92 (7 %)	Total pins	6 / 92 (7 %)
Total virtual pins	0	Total virtual pins	0
Total memory bits	0 / 276,480 (0 %)	Total memory bits	0 / 276,480 (0 %)
Embedded Multiplier 9-bit elements	0 / 30 (0 %)	Embedded Multiplier 9-bit elements	0 / 30 (0 %)
Total PLLs	0 / 2 (0 %)	Total PLLs	0 / 2 (0 %)

Рисунок 30 – Оценка затрачиваемых ресурсов (слева при разработке счетчика посредством стандартных блоков САПР, справа использования SystemVerilog)

На основе данных оценок студент должен привести обоснование и выбрать FPGA на котором будет выполняться его проект, привести его технические характеристики. Проанализировать какой из способов разработки более удобен и затрачивает меньше ресурсов FPGA.

### 3.5. Назначение входов и выходов разработанной схемы физическим I/O портам выбранной FPGA

В меню **Assignments** ⇒ **Device** следует указать выбранный в 3-м разделе КП FPGA. Не смотря на то, что САПР может само назначать функции выводам, в задачу входит назначение входов и выходов в ручную. Для этого следует проследовать в меню по пути **Assignments** ⇒ **Pin Planner** (рис. 31):

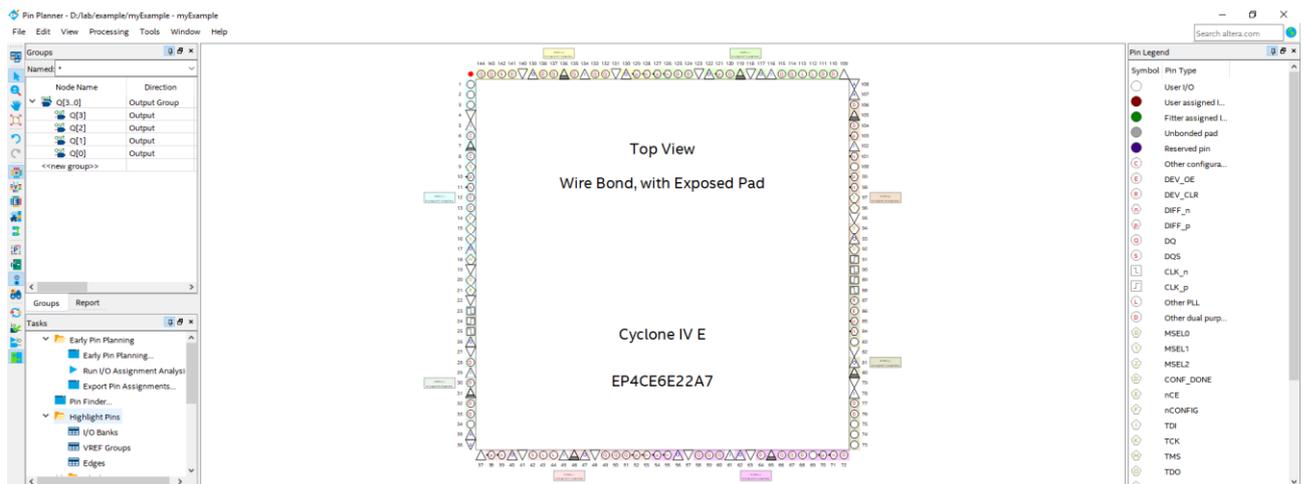


Рисунок 31 - Pin Planner

В нижней части будет приведен список входов и выходов разрабатываемой схемы:

Node Name	Direction	Location
in clk	Input	
out Q[3]	Output	
out Q[2]	Output	
out Q[1]	Output	
out Q[0]	Output	
in reset	Input	
<<new node>>		

Рисунок 32 – Список вводов-выводов проекта

В столбце **Location** нужно задать номера выводов. Для этого можно дважды кликнуть на соответствующей ячейке и выбрать номер вывода, или ввести с клавиатуры его номер. После того, как все выводы были определены (рис. 33), окно **Pin Planner** можно закрыть и перекомпилировать проект. А также проследить, чтобы после компиляции проекта не было ошибок связанных с назначением номеров портов.

Node Name	Direction	Location	I/O Bank	VREF Group	Filter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Strict Preservation
in clk	Input	PIN_24	2	B2_NO	PIN_24	2.5 V		8mA (default)			
out Q[3]	Output	PIN_73	5	B5_NO	PIN_73	2.5 V		8mA (default)	2 (default)		
out Q[2]	Output	PIN_74	5	B5_NO	PIN_74	2.5 V		8mA (default)	2 (default)		
out Q[1]	Output	PIN_75	5	B5_NO	PIN_75	2.5 V		8mA (default)	2 (default)		
out Q[0]	Output	PIN_83	5	B5_NO	PIN_83	2.5 V		8mA (default)	2 (default)		
in reset	Input	PIN_34	2	B2_NO	PIN_34	2.5 V		8mA (default)			
<<new node>>											

Рисунок 33 – Назначенный список вводов-выводов проекта

## ЛИТЕРАТУРА

1. Официальный сайт Intel FPGA and SoC [Электронный ресурс] – Режим доступа – <https://www.altera.com>
2. *Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П.* Проектирование систем на микросхемах с программируемой структурой. СПб.:БХВ-Петербург, 2006. – 736 с.
3. *Максфилд К.* Проектирование на ПЛИС. Курс молодого бойца. — М.: Издательский дом «Додэка-XX1», 2007. — 408 с.
4. *Соловьев В. В.* Основы языка проектирования цифровой аппаратуры Verilog. – М.: Горячая линия – Телеком, 2014. – 208 с.
5. *Харрис Д., Харрис С.* Цифровая схемотехника и архитектура компьютера. – Burlington:Morgan-Kaufman, 2013 – 1621 с.
6. *Злобин В.К., Григорьев В.А.* Программирование арифметических операций в микропроцессорах: Учеб.пособие для вузов. М.: Высш.шк., 1991. 302 с.
7. *Каган Б.М.* Электронные вычислительные машины и системы: Учеб. пособие для вузов . 3-е изд., перераб. и доп. М.: Энергоатомиздат, 1991. 592 с.
8. *Лысиков Б.Г.* Арифметические и логические основы цифровых автоматов. М.: Высш. шк., 1980. 336 с.
9. *Курсанов О. И., Кузнецова А. А., Аксенов А. И.* Синтез и компьютерный анализ элементов и узлов ЦВМ на базе программного пакета Micro-Cap10: методические указания по курсовому проектированию / С.-Петербург. гос. ун-т аэрокосм. приборостроения - СПб. : Изд-во ГУАП, 2015. - 32 с.

## СОДЕРЖАНИЕ

1. ОСНОВНЫЕ ПОЛОЖЕНИЯ .....	3
1.1. Цель курсового проектирования.....	3
1.2. Задание на курсовое проектирование и содержание проекта.....	3
1.3. Требования к отчету .....	4
1.4. Содержание пояснительной записки.....	4
2. ВАРИАНТЫ ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ .....	6
3. РЕКОМЕНДАЦИИ ПО МОДЕЛИРОВАНИЮ И РАЗРАБОТКЕ.....	14
3.1. Синтез схемы в САПР Quartus с использованием стандартных блоков .....	14
3.2. Пример создания собственных блоков для САПР Quartus .....	23
3.3. Синтез схемы в САПР Quartus с использованием языка описания аппаратуры SystemVerilog.....	25
3.4. Анализ затрачиваемых ресурсов и выбор FPGA .....	27
3.5. Назначение входов и выходов разработанной схемы физическим I/O портам выбранной FPGA.....	28
ЛИТЕРАТУРА .....	30