

## Упражнение 4

**Цель упражнения:**

Знакомство с системой моделирования пакета QuartusII, знакомство с процедурой создания компонентов и иерархического проекта.

**Алгоритм работы проекта:**

- Проект обеспечивает отображение на 7-сегментном индикаторе четырех разрядов:
  - Разряд 0: число 0
  - Разряд 1: число, задаваемое переключателями sw[3..0]
  - Разряд 2: число, задаваемое переключателями sw[7..4]
  - Разряд 3: число F
- Режим отображения – динамический.

## Часть 1 – Создание проекта

### 1. Проект:

- Рабочая папка – ... \ lab4;
- Имя проекта – lab4;
- Имя модуля верхнего уровня в иерархии проекта –lab4;
- СБИС ПЛ – *EP4C6E22C8*;

## Часть 2 - Создание элементов

### 1. Создайте преобразователь 2-го кода в 7-сегментный код. Имя - bin\_7seg:

- Откройте файл bin\_7seg.vhd, находящийся в папке проекта
- Создайте символ: **File=>Create Update=> Create Symbol Files for Current File**

### 2. Создайте счетчик cnt\_div\_ss:

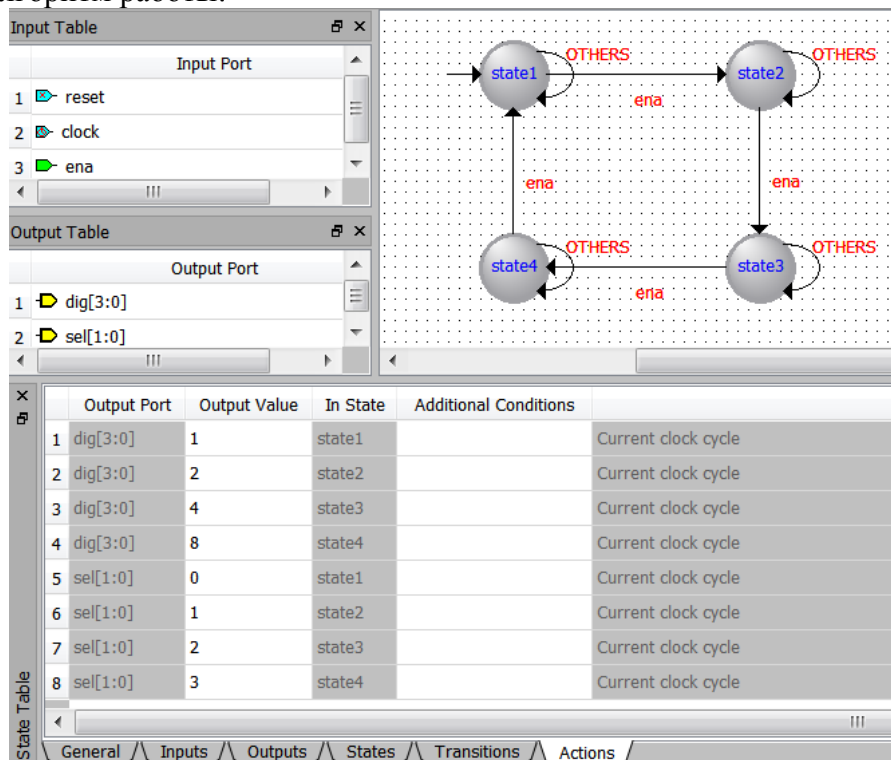
- Разрядность: 12;
- Выход: переноса (carry out)
- Модуль счета: 4

### 3. Создайте мультиплексор mux4\_1:

- Число входов: 4
- Разрядность каждого входа: 4;

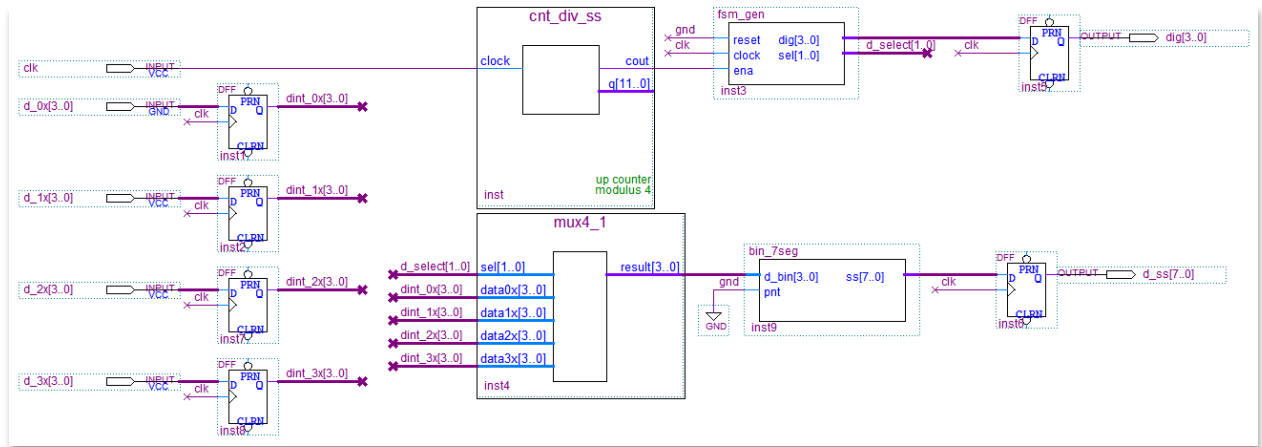
### 4. Создайте конечный автомат – распределитель импульсов. Имя - fsm\_gen:

- Вход сброса: асинхронный, активный уровень -1.
- Алгоритм работы:



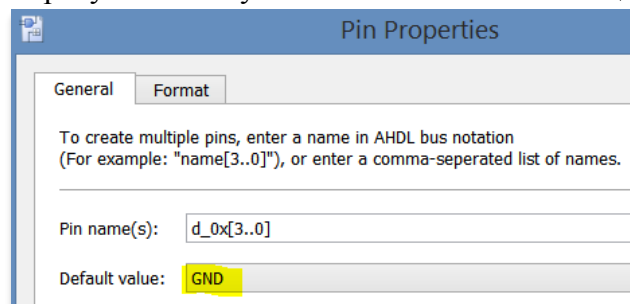
## Часть 3 - Создание схемы компонента

1. Создайте схему, изображенную на рисунке ниже.



На первом этапе создается компонент, предназначенный для иерархического проектирования. Все входы и выходы компонента должны быть синхронизированными.

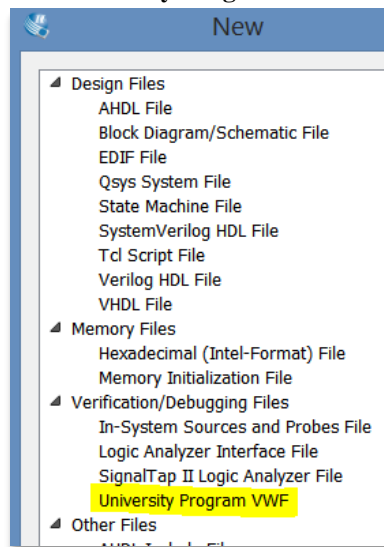
2. Для входа `d_0x[3..0]` задайте значение по умолчанию – GND
  - a. Выберите вывод
  - b. Нажмите правую клавишу мыши и выполните команду Properties



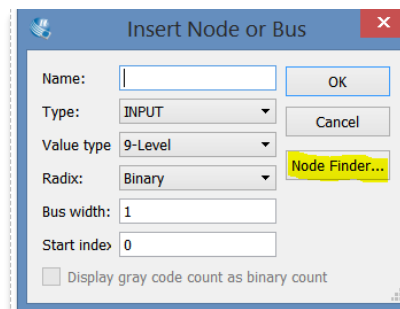
3. Проверьте, что для входа `d_3x[3..0]` по умолчанию задано значение – VCC
4. Сохраните схему под именем - `ss_cntr.bdf`.
5. На закладке **Files** навигатора проекта **Project Navigator** найдите файл `ss_cntr.bdf`, выберите его, нажмите правую клавишу мыши и выберите команду **Set as a Top-level Entity**. Созданная схема компонента станет верхним уровнем в описании.
6. Выполните команду: меню **Processing=>Start=>Start Analysis and Synthesis**.

## Часть 4 - Функциональное моделирование компонента

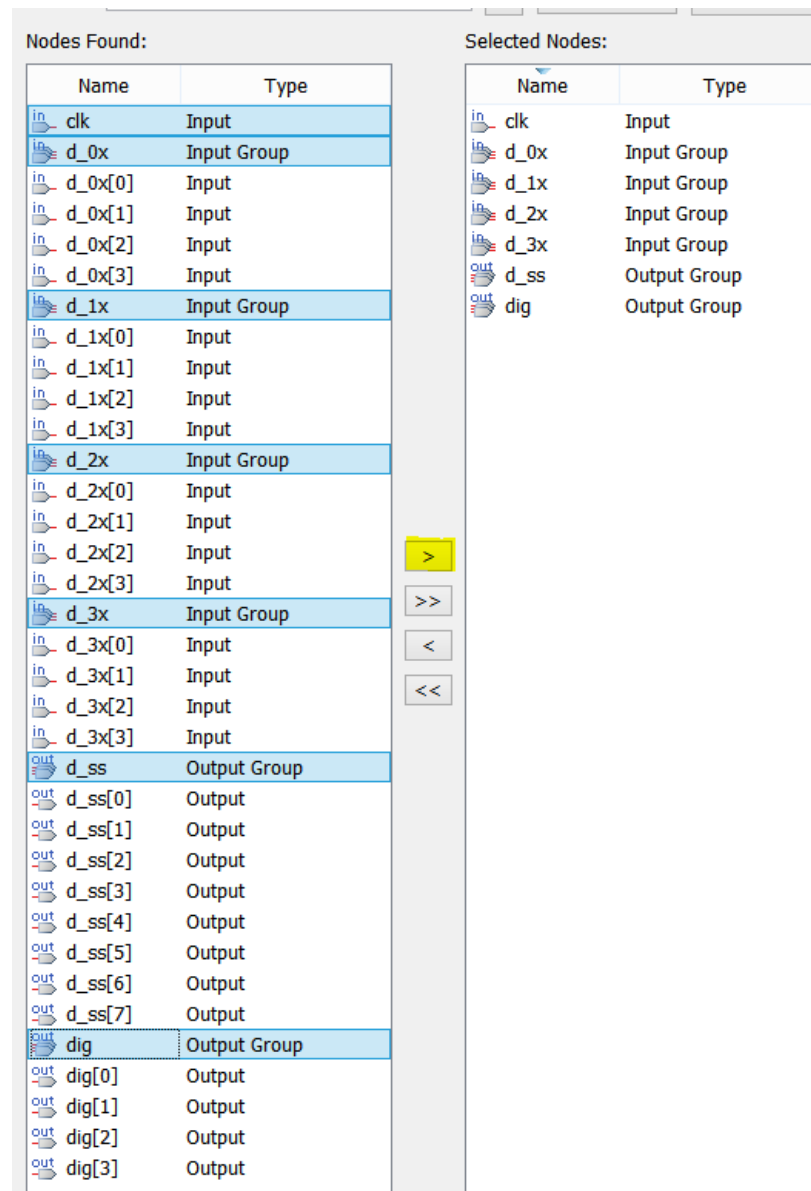
1. Выполните команду **File=>New => University Program VWF**




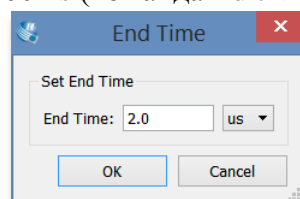
2. Откроется окно редактора тестовых воздействий
3. Для выбора выводов проекта выполните команду **Edit=>Insert=> Insert Node or Bus** и в появившемся окне запустите **Node Finder**



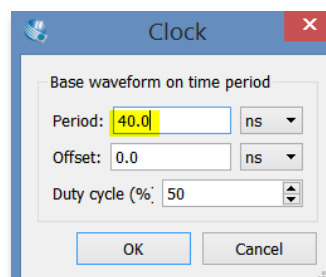
4. Откроется окно **Node Finder**, в котором следует установить фильтр (Filter) – **Pins:all** (все выводы) и нажать кнопку **List**
5. В разделе **Nodes Found** выделить интересующие сигналы и шины




6. Нажмите символ  - перенести выбранные сигналы в окно selected Nodes
7. Нажмите кнопку ОК, затем нажмите кнопку ОК еще раз.
8. Задайте длину теста равной 2000 ns (команда Edit=>Set End Time)

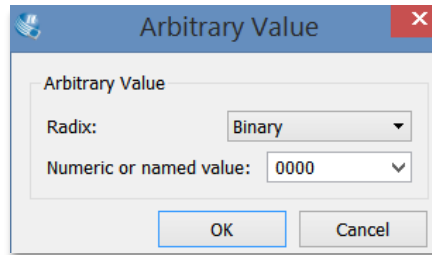


9. Введите тестовые воздействия:
  - Тактовому сигналу (выделите тактовый сигнал и выполните команду ):

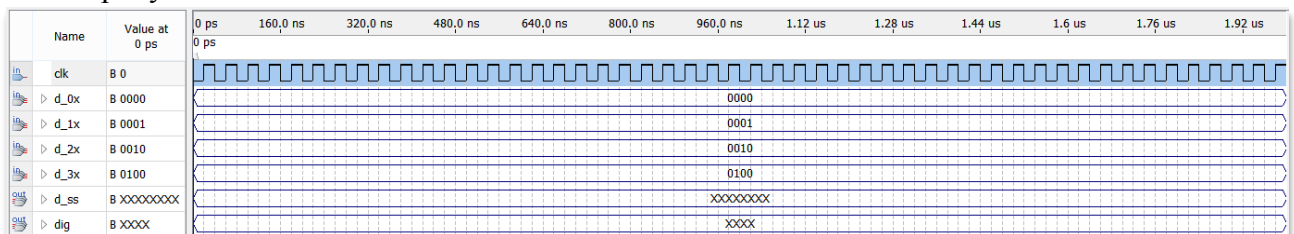


- Задайте константные значения группам сигналов (выделите группу и выполните команду ):

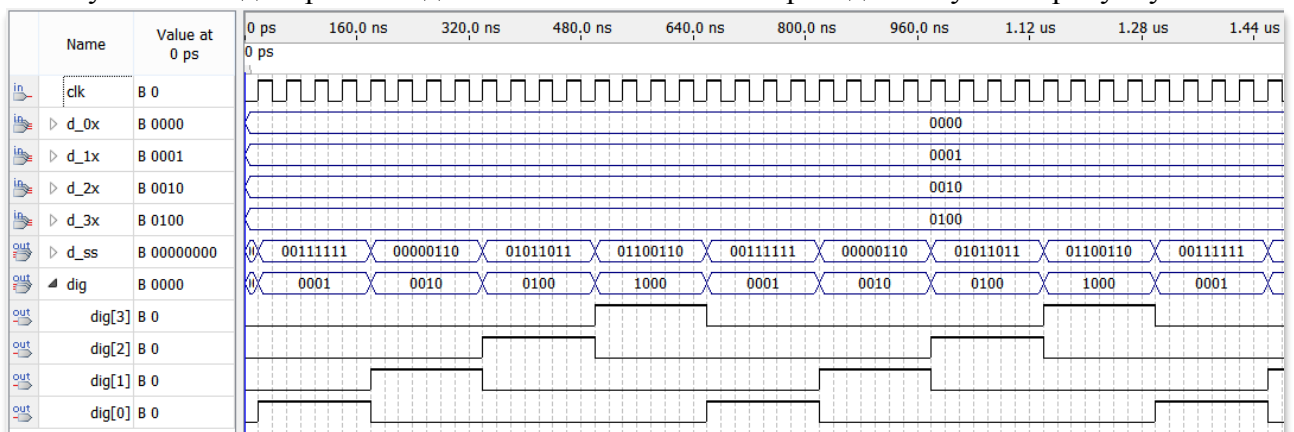
- d\_0x[3..0] – 0000
- d\_1x[3..0] – 0001
- d\_2x[3..0] – 0010
- d\_3x[3..0] – 0100



- Получившийся у Вас тест должен соответствовать тесту, приведенному на рисунке ниже



10. Сохраните тест: File=>Save (или Save as) под именем **ss\_cntr.vwf**.
11. Запустите функциональное моделирование: Simulation => Run Functional Simulation.  
Временная диаграмма с результатами моделирования откроется в отдельном окне
12. Результаты моделирования должны соответствовать приведенному ниже рисунку.

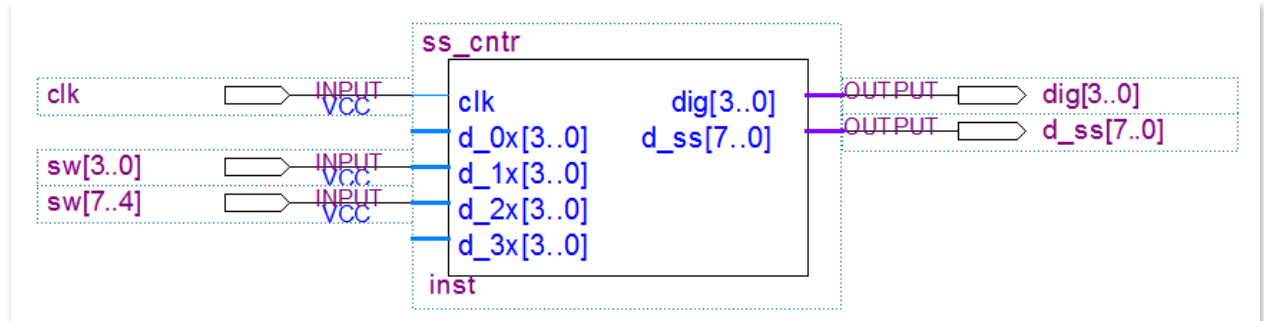


13. Объясните полученные результаты моделирования.

*Моделирование компонента закончено.*

## Часть 5 – Создание описания верхнего уровня иерархии

1. Создание символа:
  - Откройте схему **ss\_cntr.bdf**.
  - Выполните команду **File=>Create Update=> Create Symbol Files for Current File**
2. Создайте новый bdf файл.
3. Введите схему, приведенную на рисунке ниже



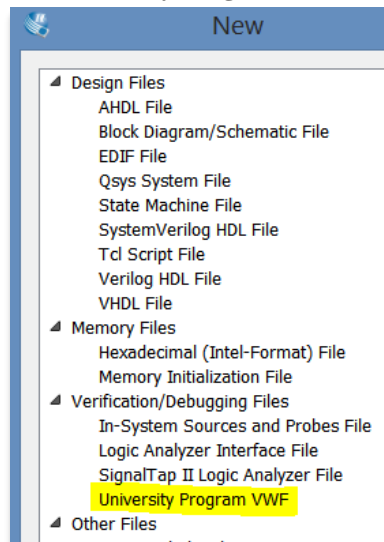
Выходы  $d\_0x[3..0]$  и  $d\_3x[3..0]$  не подключены. Ожидается, что на них будет автоматически подано 0000 и 1111 соответственно.

4. Сохраните схему под именем lab4.bdf.
5. Выполните команду: **Project=> Set as a Top-level Entity** - файл станет файлом верхнего уровня в иерархии описаний проекта.
6. Осуществите компиляцию проекта: **Processing=>Start=>Start Analysis and Synthesis**.



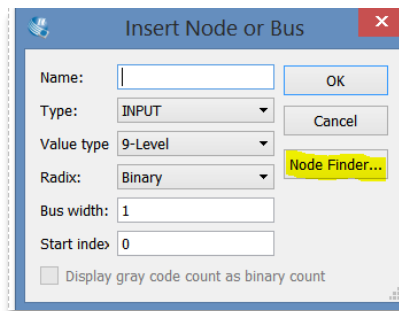
## Часть 6 - Функциональное моделирование проекта

1. Выполните команду **File=>New => University Program VWF**

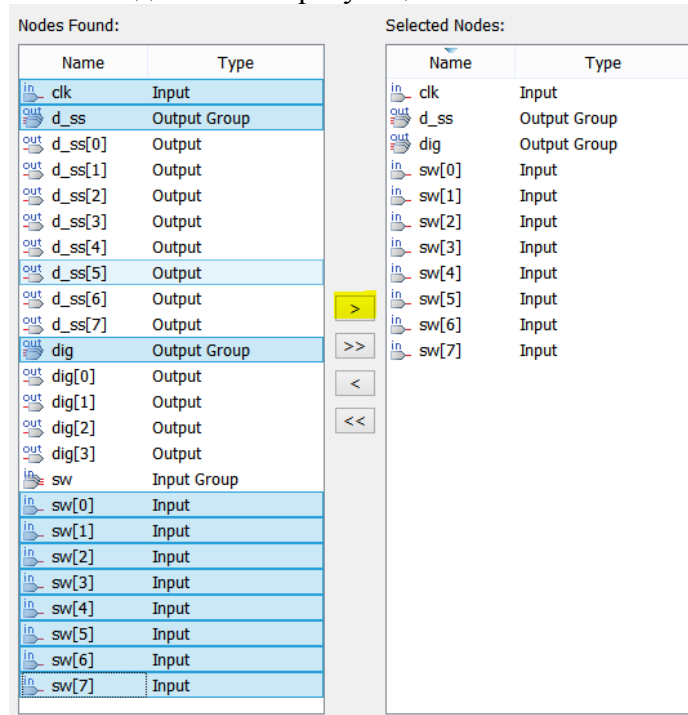



2. Откроется окно редактора тестовых воздействий
3. Для выбора выводов проекта выполните команду **Edit=>Insert=> Insert Node or Bus** и в появившемся окне запустите Node Finder





4. Откроется окно Node Finder, в котором следует установить фильтр (Filter) – Pins:all (все выводы) и нажать кнопку List
5. В разделе Nodes Found выделите интересующие сигналы и шины



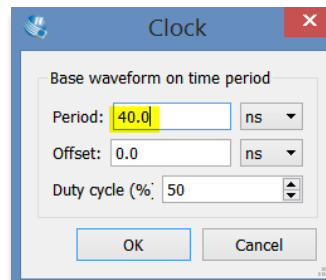
6. Нажмите символ  - перенести выбранные сигналы в окно selected Nodes
7. Нажмите кнопку OK, затем нажмите кнопку OK еще раз.
8. В окне редактора тестовых воздействий переместите и сгруппируйте выходы так, как показано на рисунке ниже (в группе sw\_a – MSB должен быть SW[3], а в группе sw\_b – MSB должен быть sw[7]), для отображения данных установите систему счисления Hexadecimal (для всех групп)


	Name	Value at 0 ps
in	clk	B 0
in	sw_a	B 0000
in	sw[3]	B 0
in	sw[2]	B 0
in	sw[1]	B 0
in	sw[0]	B 0
in	sw_b	B 0000
in	sw[7]	B 0
in	sw[6]	B 0
in	sw[5]	B 0
in	sw[4]	B 0
out	dig	B XXXX
out	d_ss	B XXXXXXXX

9. Задайте длину теста равной 1000 ns (команда Edit=>Set End Time)

10. Введите тестовые воздействия:

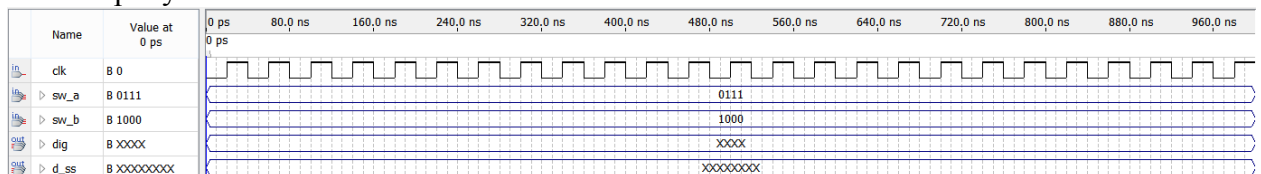
- Тактовому сигналу (выделите тактовый сигнал и выполните команду ):



- Задайте константные значения группам сигналов (выделите группу и выполните команду ):

- sw\_a – 0111
- sw\_b – 0001

- Получившийся у Вас тест должен соответствовать тесту, приведенному на рисунке ниже

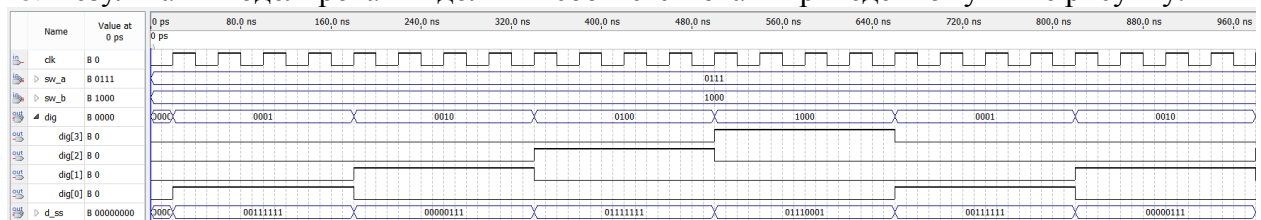


11. Сохраните тест: File=>Save (или Save as) под именем **lab4.vwf**.

12. Запустите функциональное моделирование: Simulation => Run Functional Simulation.

Временная диаграмма с результатами моделирования откроется в отдельном окне

13. Результаты моделирования должны соответствовать приведенному ниже рисунку.

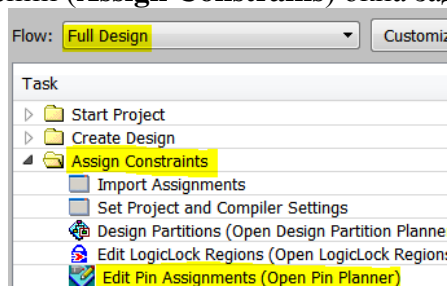


14. Объясните полученные результаты моделирования.

Моделирование закончено.

## Часть 7 – Назначение контактов СБИС

1. Запустите редактор назначений контактов СБИС: команда **Edit Pin Assignments (Open Pin Planer)** в папке назначений (**Assign Constrains**) окна задач (**Task**)



2. Назначьте контакты СБИС выводам проекта и используемый стандарт сигналов в соответствии с приведенной ниже таблицей

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Current Strength	Slew Rate
clk	Input	PIN_23	1	B1_NO	3.3-V LVCMOS	2mA (default)	
d_ss[7]	Output	PIN_75	5	B5_NO	3.3-V LVCMOS	2mA (default)	2 (default)
d_ss[6]	Output	PIN_84	5	B5_NO	3.3-V LVCMOS	2mA (default)	2 (default)
d_ss[5]	Output	PIN_76	5	B5_NO	3.3-V LVCMOS	2mA (default)	2 (default)
d_ss[4]	Output	PIN_85	5	B5_NO	3.3-V LVCMOS	2mA (default)	2 (default)
d_ss[3]	Output	PIN_77	5	B5_NO	3.3-V LVCMOS	2mA (default)	2 (default)
d_ss[2]	Output	PIN_86	5	B5_NO	3.3-V LVCMOS	2mA (default)	2 (default)
d_ss[1]	Output	PIN_133	8	B8_NO	3.3-V LVCMOS	2mA (default)	2 (default)
d_ss[0]	Output	PIN_87	5	B5_NO	3.3-V LVCMOS	2mA (default)	2 (default)
d_iq[3]	Output	PIN_73	5	B5_NO	3.3-V LVCMOS	2mA (default)	2 (default)
d_iq[2]	Output	PIN_80	5	B5_NO	3.3-V LVCMOS	2mA (default)	2 (default)
d_iq[1]	Output	PIN_74	5	B5_NO	3.3-V LVCMOS	2mA (default)	2 (default)
d_iq[0]	Output	PIN_83	5	B5_NO	3.3-V LVCMOS	2mA (default)	2 (default)
sw[7]	Input	PIN_88	5	B5_NO	3.3-V LVTTTL	8mA (default)	
sw[6]	Input	PIN_89	5	B5_NO	3.3-V LVTTTL	8mA (default)	
sw[5]	Input	PIN_90	6	B6_NO	3.3-V LVTTTL	8mA (default)	
sw[4]	Input	PIN_91	6	B6_NO	3.3-V LVTTTL	8mA (default)	
sw[3]	Input	PIN_49	3	B3_NO	3.3-V LVTTTL	8mA (default)	
sw[2]	Input	PIN_46	3	B3_NO	3.3-V LVTTTL	8mA (default)	
sw[1]	Input	PIN_25	2	B2_NO	3.3-V LVTTTL	8mA (default)	
sw[0]	Input	PIN_24	2	B2_NO	3.3-V LVTTTL	8mA (default)	

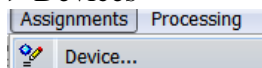
Назначение контакта можно выполнить путем ввода номера контакта в поле location; или перетаскиванием вывода на контакт СБИС.

Назначение стандарта можно выполнить путем выбора стандарта из списка в ячейке столбца I/O Standard для каждого вывода отдельно; или задания стандарта группе выводов.

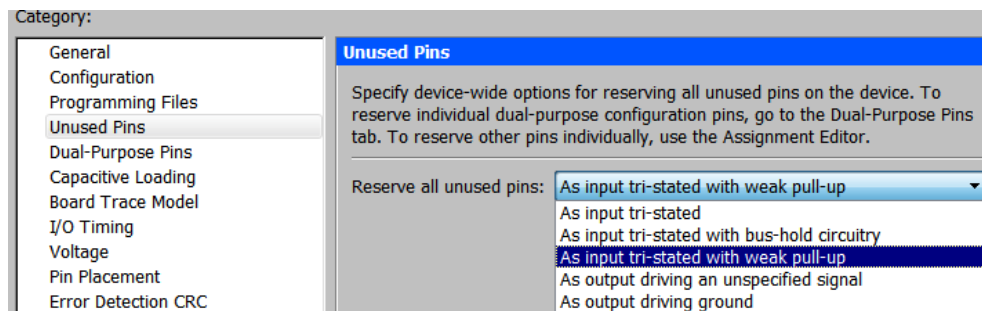
3. Закройте редактор назначения контактов.

## Часть 8 – Задание режима работы неиспользуемых контактов СБИС

1. Выполните команду Assignments=> Devices



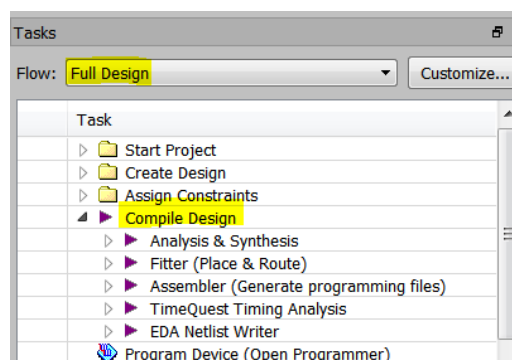
2. В окне Device нажмите кнопку Device and Pin Options
3. В окне Device and Pin Options перейдите к категории Unused Pins и выберите режим **As input tri-stated with weak pull-up** для всех не использованных контактов СБИС



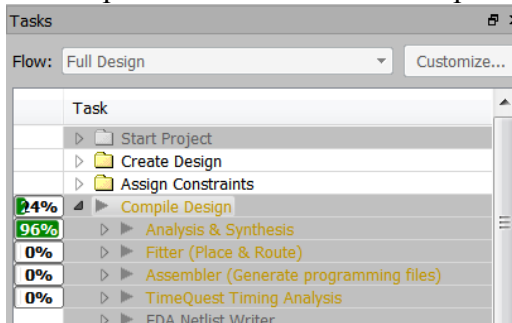
4. Нажмите кнопку ОК. В окне Device нажмите кнопку ОК еще раз.

## Часть 9 – Компиляция проекта

1. В окне задач (Tasks) выберите процедуру Full Design и двойным щелчком левой клавиши мыши по команде Compile Design запустите полную компиляцию проекта.



2. Окно задач (Tasks) будет отображать ход выполнения процедуры компиляции.

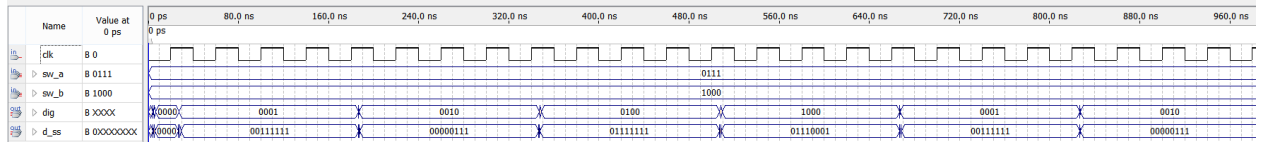


*В процессе полной компиляции проекта осуществляется:*

- ✓ проверка синтаксиса,
- ✓ синтез с оптимизацией занимаемой площади и быстродействия проекта,
- ✓ трассировка и СБИС с оптимизацией занимаемой площади и быстродействия проекта,
- ✓ получение файла для конфигурирования СБИС – *prof(sof)* файл,
- ✓ получение модели с временными параметрами реализованной СБИС,
- ✓ временной анализ,
- ✓ Формирования файла с детальным отчетом о всех этапах компиляции проекта.

## Часть 10 – Временное моделирование

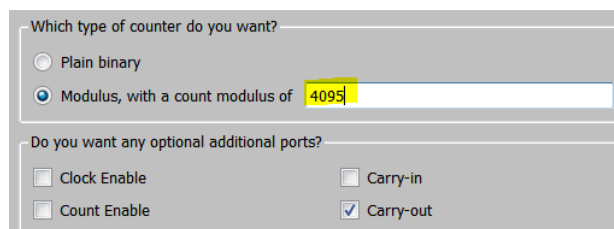
1. Откройте файл lab4.vwf
2. Запустите временное моделирование: Simulation => Run Timing Simulation. Временная диаграмма с результатами моделирования откроется в отдельном окне
3. Результаты моделирования должны соответствовать приведенному ниже рисунку.



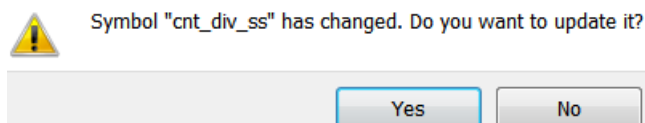
4. Объясните в чем основное отличие полученных результатов от результатов функционального моделирования, полученных ранее.

## Часть 11 – Изменение схемы для аппаратной реализации и компиляция проекта

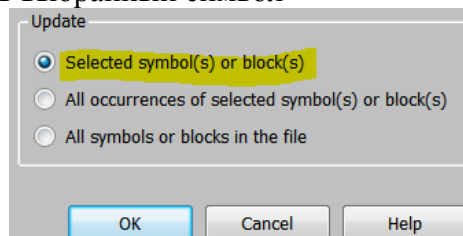
1. Для аппаратной реализации модуля динамической индикации следует изменить модуль счета счетчика cnt\_div\_ss, обеспечивающего деление входной частоты 25 МГц.
2. В пакете QII откройте схему верхнего уровня иерархии описаний – lab4.gdf, двойным щелчком по компоненту ss\_cntr перейдите на нижний уровень иерархии описания – схему компонента.
3. Двойным щелчком в зоне символа счетчика cnt\_div\_ss запустите помощник MegaWizard Plug-in Manager
4. В окне настроек счетчика перейдите на страницу 2 и задайте модуль счета равным 4095



5. Нажмите кнопку Finish, затем нажмите ее еще раз.
6. В появившемся окне нажмите OK
7. В следующем окне нажмите кнопку YES



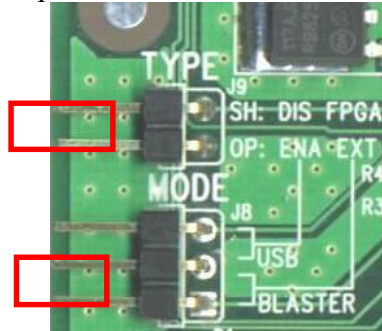
8. Далее укажите: обновить выбранный символ



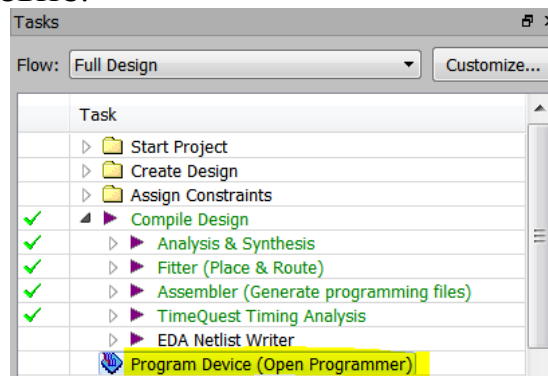
9. Проверьте в схемном редакторе, что символ счетчика обновлен и все выводы подключены правильно.
10. Сохраните схему компонента.
11. Осуществите полную компиляцию проекта.

## Часть 12 – Конфигурирование СБИС и проверка проекта на плате

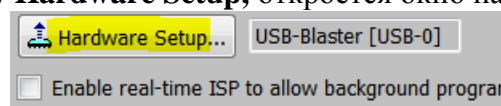
1. На плате miniDiLaB-CIV установите джамперы следующим образом:
  - a. Соедините выводы разъема “TYPE”
  - b. Соедините выводы 1-2 разъема “MODE”



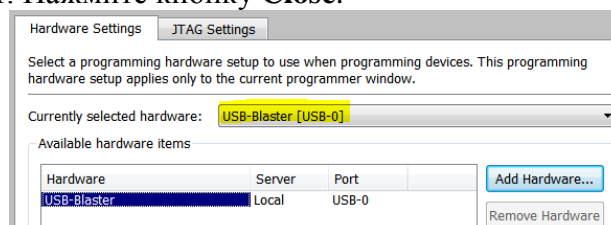
2. Подсоедините входящий в комплект поставки USB кабель A-miniB к USB 2.0 порту компьютера (должен обеспечивать ток до 500мА), а затем к плате miniDiLaB-CIV.
3. Включите плату miniDiLaB-CIV : переключатель Power
4. В окне задач (Tasks) выберите процедуру Full Design и двойным щелчком левой клавиши мыши по команде Program Device запустите приложение, управляющее конфигурированием СБИС.



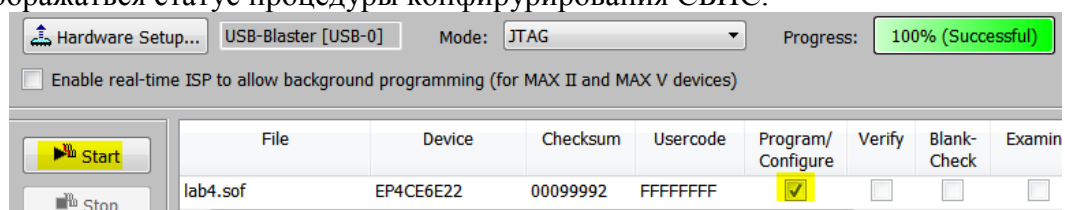
5. Откроется окно управления конфигурированием СБИС.
6. Для установки интегрированного на плату miniDiLaB-CIV средства конфигурирования СБИС нажмите кнопку **Hardware Setup**, откроется окно настроек.



7. В разделе **Available hardware items** выберите (двойным щелчком левой клавиши мыши) USB-Blaster. Нажмите кнопку **Close**.

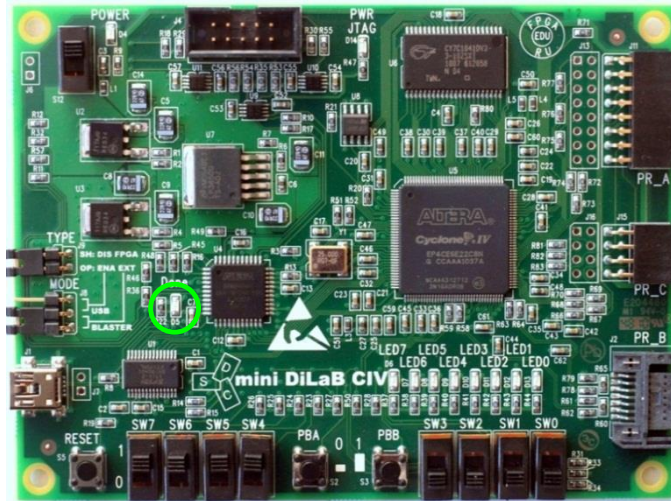


8. Включите опцию **Program/Configure** и нажмите кнопку **Start**. В окне Progress будет отображаться статус процедуры конфигурирования СБИС.





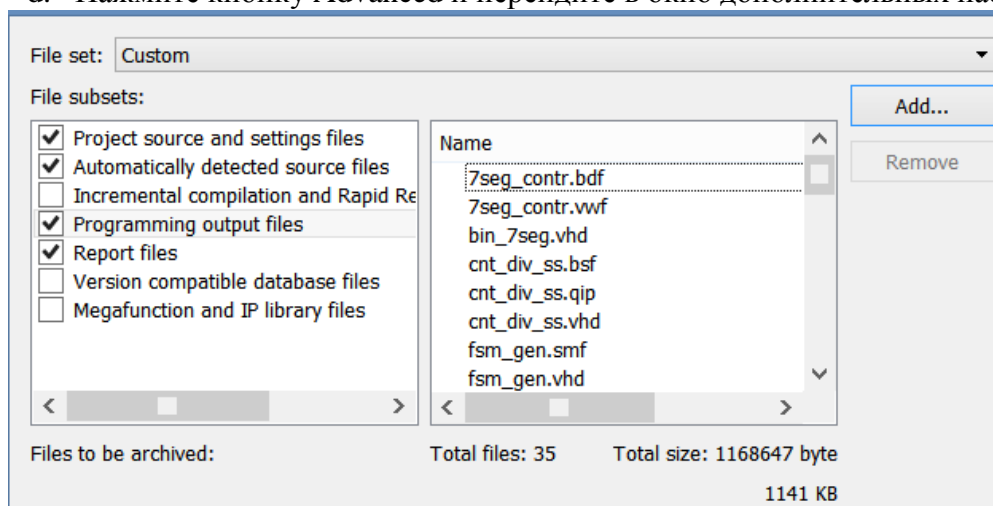
9. Когда СБИС будет запрограммирована на плате miniDiLaB-CIV загорится зеленый светодиод – “Done”.



10. Проверьте работу проекта:
- Разряд 3 7-сегментного индикатора отображает F
  - Разряд 2 7-сегментного индикатора отображает 16-ричный код числа, установленного переключателями sw[7..4]
  - Разряд 1 7-сегментного индикатора отображает 16-ричный код числа, установленного переключателями sw[3..0]

## Часть 13 – Архивация и восстановление проекта

- Создайте архив проекта:
  - В рабочей папке проекта (...\\lab4) создайте папку archives
  - Выполните команду Project=> Archive Project
  - В появившемся окне укажите папку для архива - archives и задайте имя архива lab4
  - Нажмите кнопку Advanced и перейдите в окно дополнительных настроек



- Выберите опции, показанные на рисунке выше, зафиксируйте объем включенных в архив файлов.
- Нажмите ОК и затем запустите процедуру архивации.
- Найдите в папке archives созданные 2 файла: lab4.qar и lab4.qarlog. Определите объем архива, сравните его с объемом файлов, включенных в архив

2. Восстановление архива
  - a. В папке проекта (...\\lab4) создайте папку restored
  - b. Выполните команду Project=> Restore Archived Project
  - c. В появившемся окне с помощью браузера найдите архив lab4.qar и укажите папку распаковки архива (...\\lab4\\restored)
  - d. Нажмите кнопку ОК. Архив будет восстановлен и будет открыт восстановленный проект.
3. Откройте программатор и осуществите программирование СБИС (убедитесь, что файлы для программирования были помещены в архив).

*Лабораторная работа завершена.*