

## Лабораторная работа 4

### Изучение IP-компонентов, редактора назначений и анализ реализации проекта в САПР Quartus Prime Lite

**Цель лабораторной работы:** Знакомство с базовыми возможностями схемного редактора (использование IP-компонентов) и редактора назначений САПР Quartus Prime. Подключение внешних библиотек и анализ реализации проекта средствами САПР Quartus Prime на всех этапах проектирования.

Работа включает следующие этапы:

- Создание проекта
- Подключение внешней библиотеки
- Создание файла инициализации памяти ROM, реализующего функцию преобразователя 16-разрядного двоичного кода в 12-разрядный двоично-десятичный код
- Создание экземпляра модуля памяти ROM на базе библиотеки IP-компонентов
- Создание экземпляра умножителя на базе библиотеки IP-компонентов
- Создание экземпляра умножителя тактовой частоты и блока управления глобальным тактовым сигналом на базе библиотеки IP-компонентов
- Создание схемы проекта на базе созданных компонентов
- Функциональное моделирование проекта.
- Анализ проекта с помощью RTL-Viewer, Technology Map Viewer (Post-Mapping)
- Назначение выводов ПЛИС
- Создание sdc – файла и выполнение полной компиляции проекта
- Анализ проекта с помощью Technology Map Viewer (Post-Fitting) и State Machine Viewer
- Анализ проекта с помощью Chip Planner
- Управление проектом с помощью Assignment Editor (редактор назначений)
- Временное моделирование проекта
- Изменение параметров схемы проекта и его повторная полная компиляция
- Конфигурирование ПЛИС и проверка работоспособности на отладочной плате.

**Алгоритм работы проекта:** В проекте реализована схема динамической индикации информации на светодиодном 7-сегментном 4-х разрядном индикаторе. На 3 разрядах индикатора отображается результат умножения двух 4-разрядных чисел:

числа А, которое задается с помощью ползунковых переключателей SW7...SW4;  
и числа В, которое задается с помощью ползунковых переключателей SW3...SW0.

#### 1. Создание проекта

##### 1.1. Запустите пакет проектирования Quartus Prime Lite.

**Замечание.** Порядок создания проекта подробно описан в документе «Практическое занятие 1\_v1».

1.2. Для создания проекта выполните следующие действия:

**File => New Project Wizard.**

В появившемся окне “**Introduction**” выполните «**Next**».

1.3. В окошке “**Directory, Name, Top-Level Entity**” (рис. 1) задайте (или выберите существующую) директорию с проектами и название проекта **lab4\_<fio>** (<fio> – аббревиатура на латинице из первых букв ФИО).

**Важно!**

1. Желательно выделить папку для проектов в корневом каталоге.
2. Не используйте кириллицу и пробелы в определении путей и названий проектов и

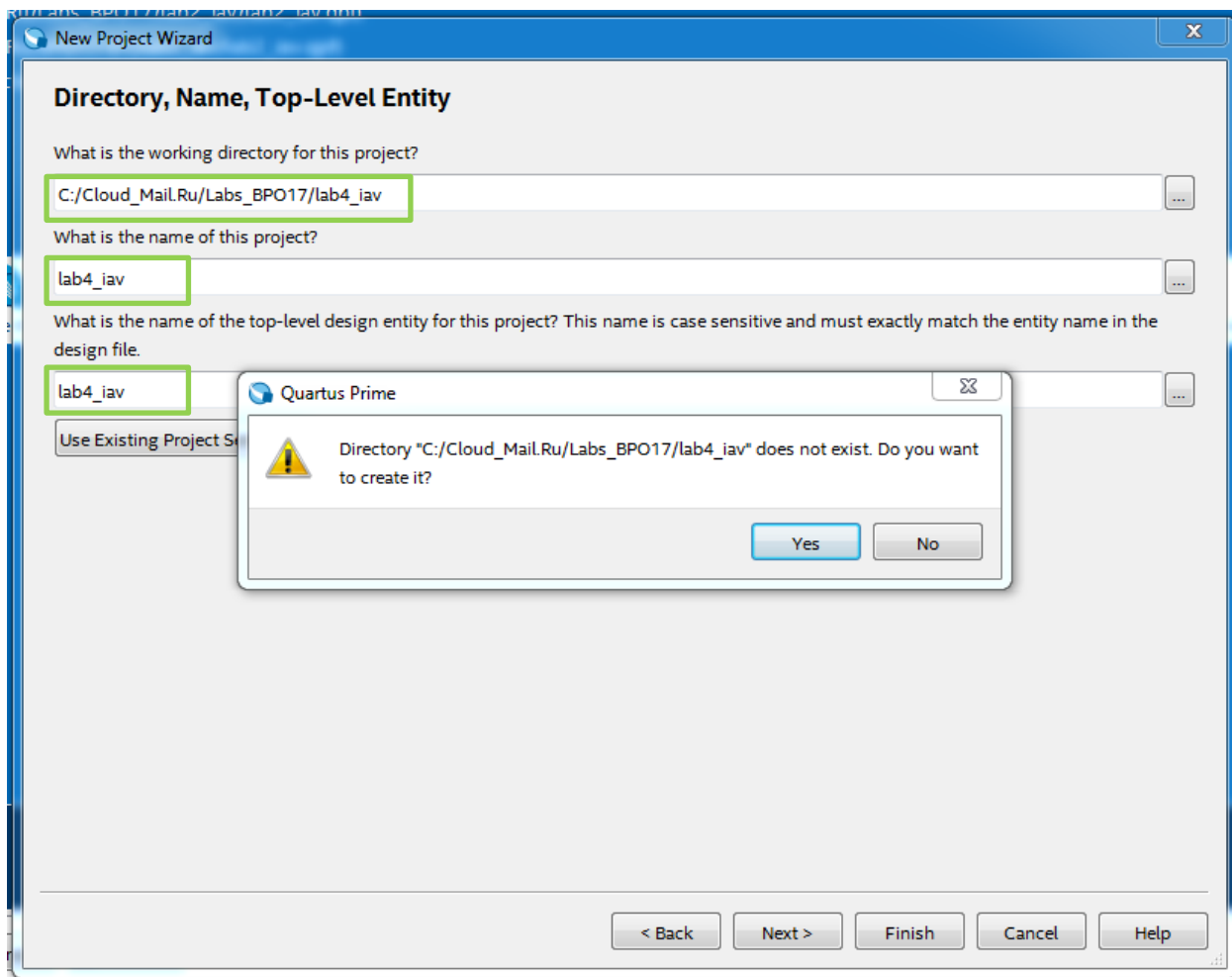


Рис.1 Окно выбора директории для установки проекта

Выполните «**Next**», далее «**Yes**» на предложение создать новую директорию для создаваемого проекта (если директория не создана ранее).

1.4. Появится окно “**Project Type**”, выберите *empty project* (пустой проект) и затем выполните «**Next**».

1.5. Появится окно “**Add Files**”, выполните «**Next**».

1.6. Появится окно “**Family, Device & Board Settings**”. На панели *Device family* выбирается семейство ПЛИС *Cyclone IVE*, а из таблицы *Available devices* нужная микросхема **EP4CE6E22C8**. Для облегчения поиска микросхемы можно конкретизировать ее параметры с помощью выбора параметров «тип корпуса» (*package*), «количество выводов» (*pin count*), «быстродействие ядра» (*core speed grade*) из соответствующих выпадающих меню панели *Show in 'Available devices' list* (рис. 2). Для проектов практических и лабораторных работ необходимо выбирать микросхему, установленную на отладочную плату, с помощью которой выполняются данные работы. Выполните «**Next**».

1.7. Появится окно “**EDA Tool Settings**”, выполните «**Next**».

1.8. Появится последнее окно создания проекта “**Summary**”, в котором можно проверить все настройки, сделанные на предыдущих шагах. Выполните «**Finish**».

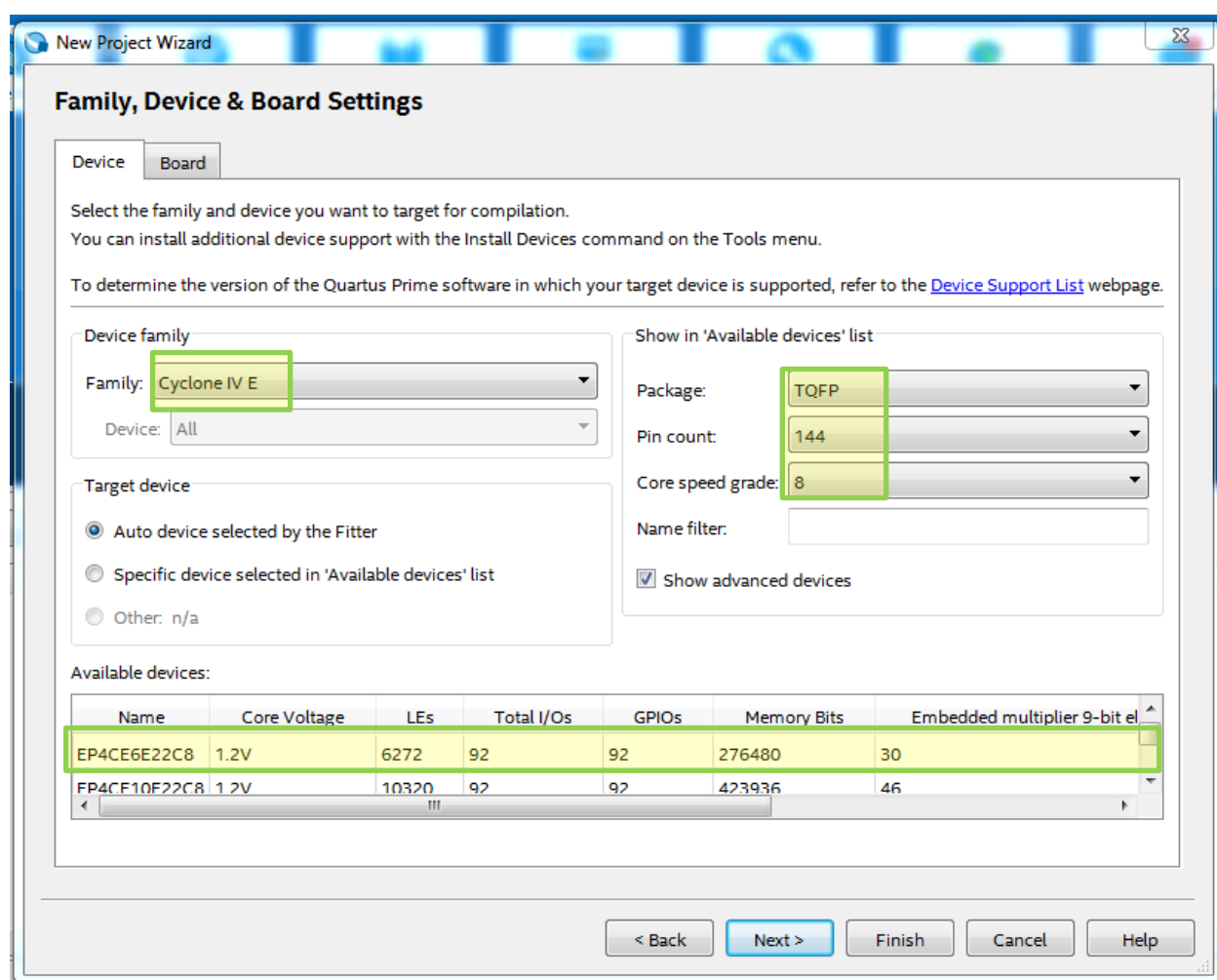


Рис.2 Окно для выбора микросхемы


После этого вновь созданный проект становится текущим в запущенной системе проектирования **Quartus Prime**.

## 2. Подключение внешней библиотеки

2.1. В данной лабораторной работе в качестве схемы управления индикатором будет использоваться компонент **ind\_drv**, созданный в лабораторной работе 3.

2.2. Выполните команду

**Assignments => Settings => Libraries**

2.3. В открывшемся окне, в строке **Project library name**, через поисковую кнопку  выберите папку с проектом **lab3\_iav** (рис. 3) и добавьте в текущий проект с помощью кнопки «Add».

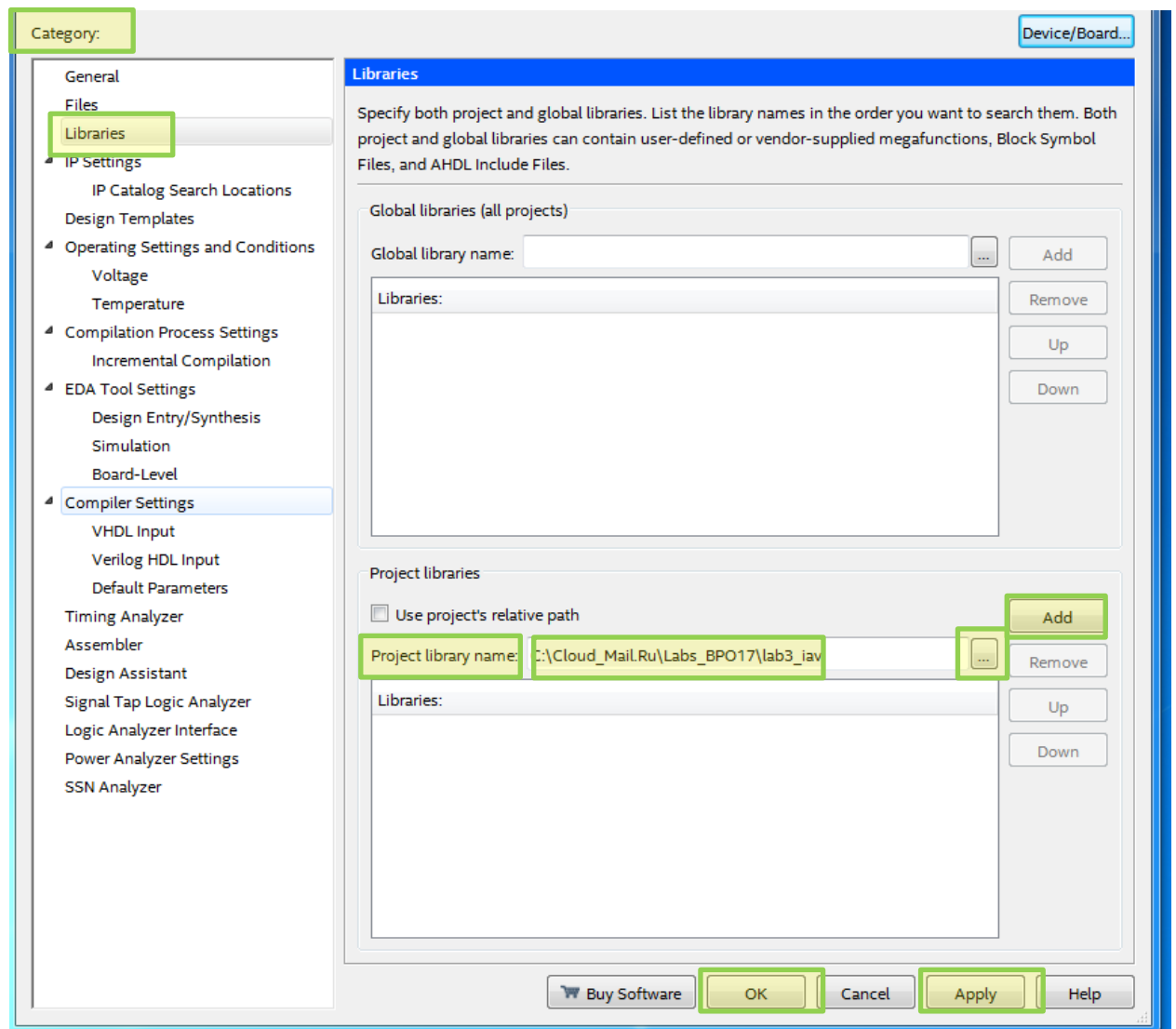


Рис.3 Окно для подключения внешней библиотеки

В поле **Libraries** появится папка с выбранным проектом **lab3\_iav** (рис. 4).

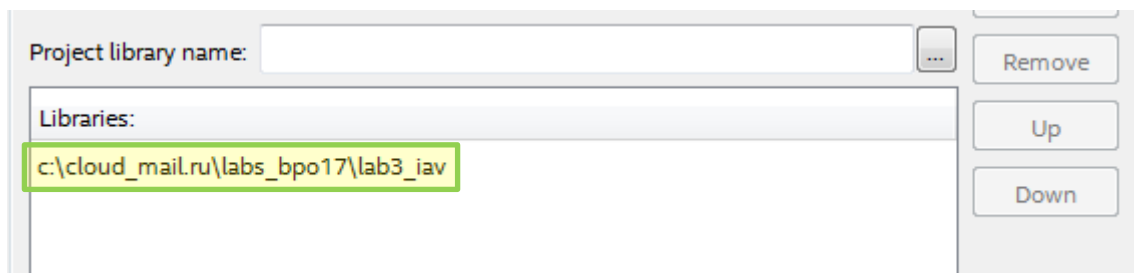


Рис.4 Окно с подключенной внешней библиотекой

2.4. Выполните «**Apply**», затем «**Ok**». Папка с проектом **lab3\_iav** подключена к текущему проекту в качестве внешней библиотеки.

### 3. Создание файла инициализации памяти ROM, реализующего функцию преобразователя 16-разрядного двоичного кода в 12-разрядный двоично-десятичный код

3.1. Выполните «**File => New**», откроется окошко «**New**» (рис.5) с выбором типов создаваемых файлов.

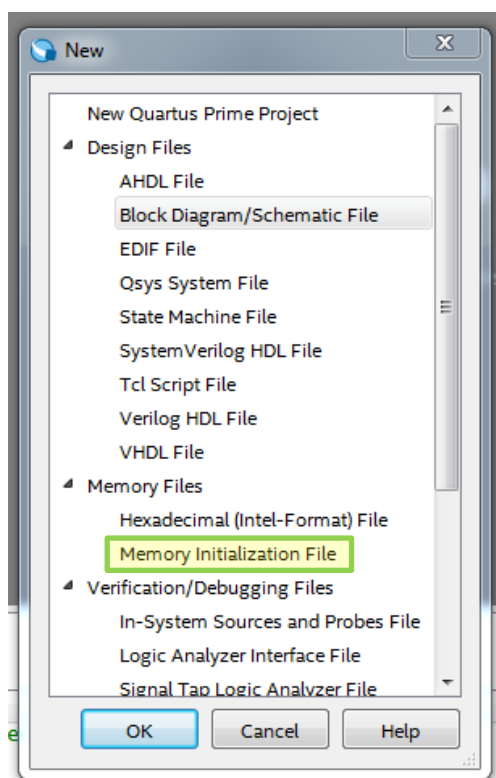


Рис. 5 Окно выбора типа создаваемого проектного файла (design file)

3.2. Выберите **Memory Files => Memory Initialization File**, далее «**OK**».

3.3. В появившемся окошке «**Number of Words & Word Size**» задайте параметры памяти ROM, для которой создается файл инициализации (рис. 6). Далее «**OK**». Откроется окно редактора памяти (рис.7). Вид окна может отличаться от показанного на рисунке. Это зависит от текущих настроек окна редактора памяти в пакете. Если вид окна редактора памяти отличается, выполните «**View=>Cells Per Row=>1**».

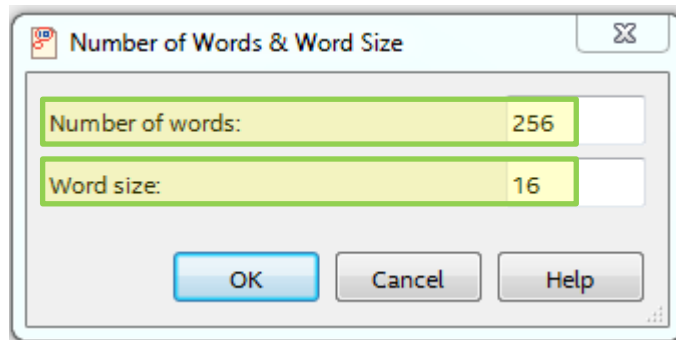


Рис. 6 Окно выбора параметров памяти

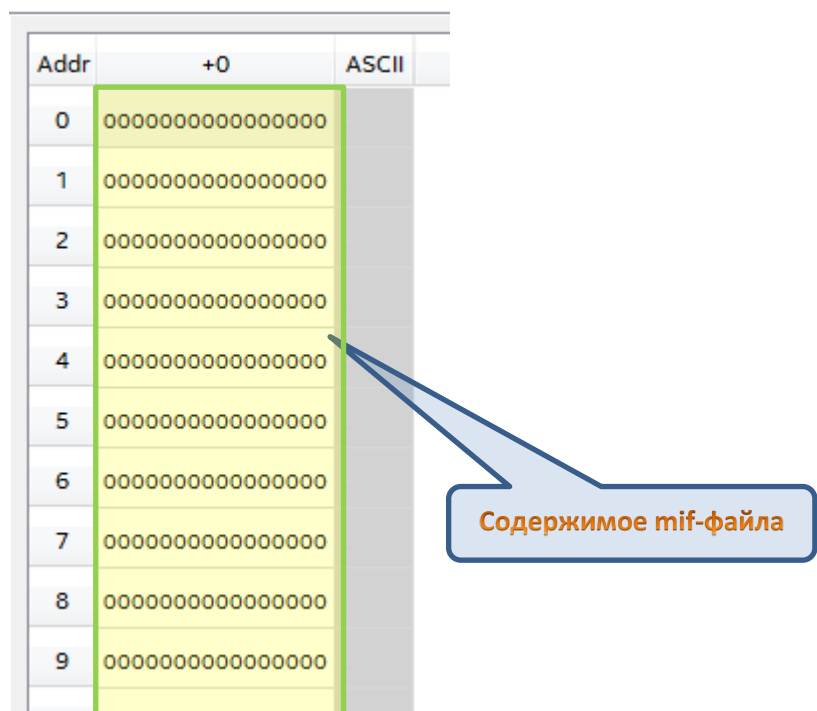


Рис. 7 Окно редактора памяти

3.4. Установите систему счисления данных в памяти в **Unsigned Decimal**

«**View=>Memory Radix=> Unsigned Decimal**»

Окно редактора памяти видоизменится следующим образом (рис. 8, **Mif1.mif**):

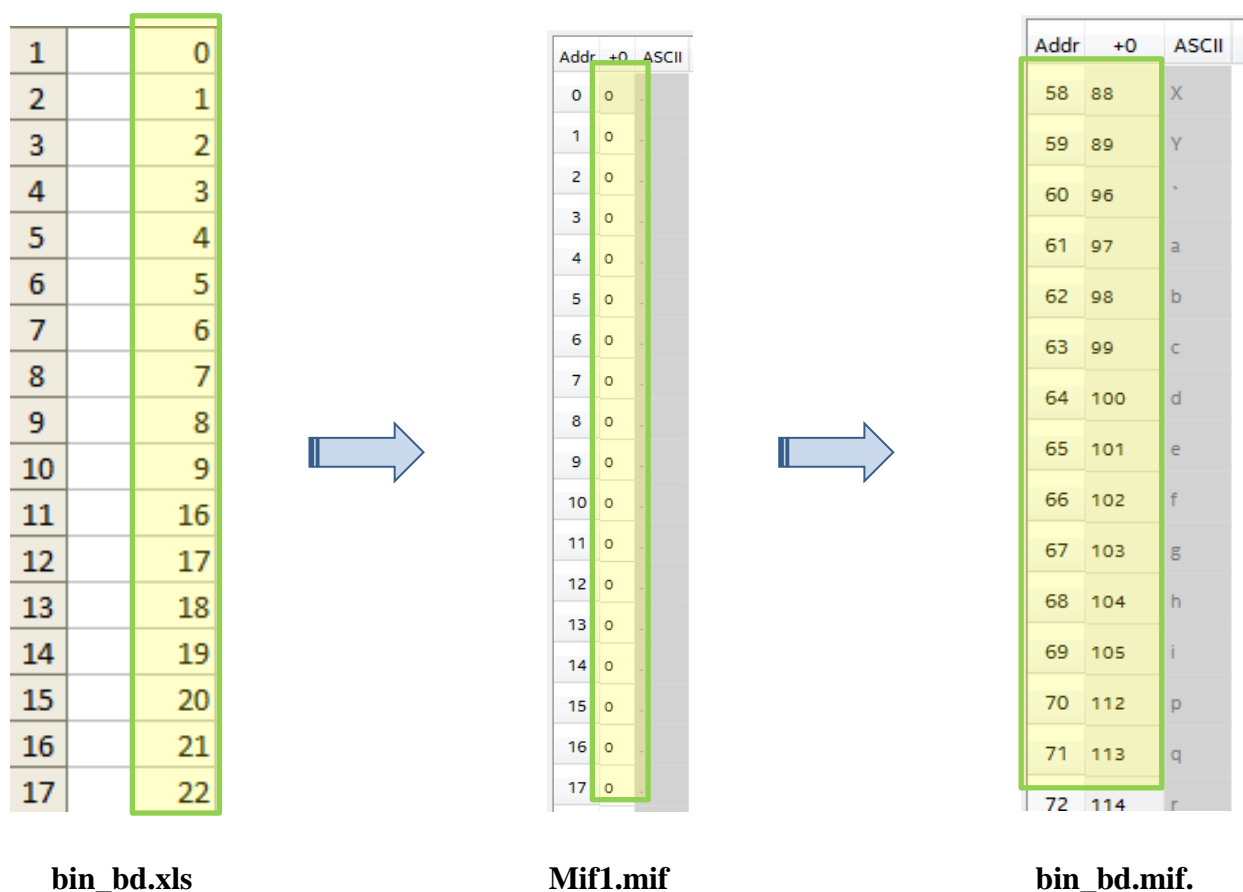


Рис. 8 Окно редактора памяти

3.5. Откройте файл **bin\_bd.xls** и скопируйте содержимое колонки «A» в созданный **Mif1.mif** – файл (рис. 8). Сохраните под именем **bin\_bd.mif**.

#### 4 Создание экземпляра модуля памяти ROM

4.1. В разделе **Library => Basic Functions => On Chip Memory** окна **IP Catalog** выберите мегафункцию **ROM: 1-PORT** (рис. 9) и нажмите кнопку «+ADD».

4.2. В появившемся окне настройки задайте имя экземпляра выбранного компонента (**ROM**) и укажите язык описания аппаратуры (VHDL), в коде которого он будет храниться в проекте (рис.10). Выполните «OK».

4.3. В появившемся окне “**MegaWized Plug-In Manager [page 1 of 5]**” установите параметры как на рис. 11. Выполните «Next».

4.4. В появившемся окне “**MegaWized Plug-In Manager [page 2 of 5]**” параметры должны быть такие же, как на рис. 12. Выполните «Next».

4.5. В появившемся окне “**MegaWized Plug-In Manager [page 3 of 5]**” установите параметры как на рис. 13. В окошке **File name** с помощью кнопки **Browse** выберите созданный ранее файл **bin\_7seg.mif**. Выполните «Open», затем «Next».

4.6. В появившемся окне “**MegaWized Plug-In Manager [page 4 of 5]**” выполните «Next».

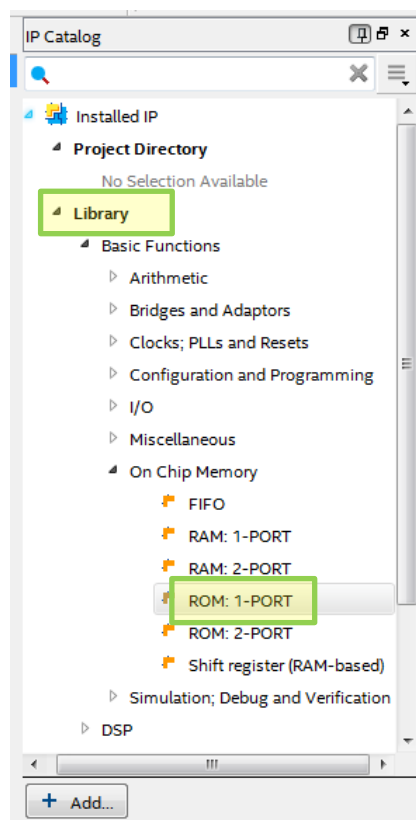


Рис. 9 Окно IP Catalog

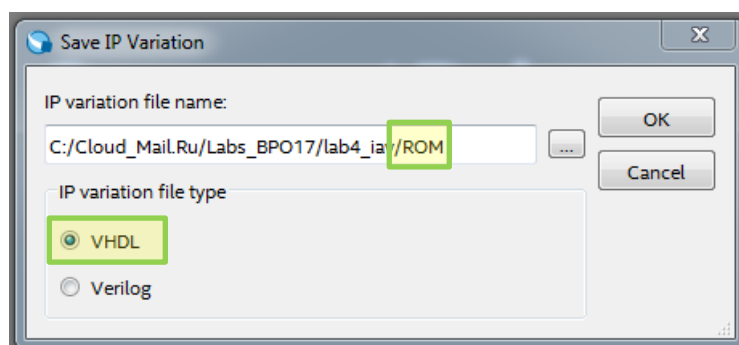


Рис. 10 Окно создания экземпляра выбранного компонента

4.7. В появившемся окне “MegaWizards Plug-In Manager [page 5 of 5]” выберите создаваемые файлы, как на рис. 14. Далее «Finish». Экземпляр компонента памяти ROM создан и его содержимое связано с файлом инициализации памяти **bin\_bd.mif**. Файл **ROM.vhd** содержит информацию о текстовом описании созданного экземпляра компонента (vhd – сокращение от названия языка описания аппаратуры VHDL) на языке VHDL, может использоваться для создания текстового описания проекта. Файл **ROM.bsf** содержит информацию о графическом обозначении созданного экземпляра компонента (block symbol file), далее будет использоваться для создания схемы проекта.



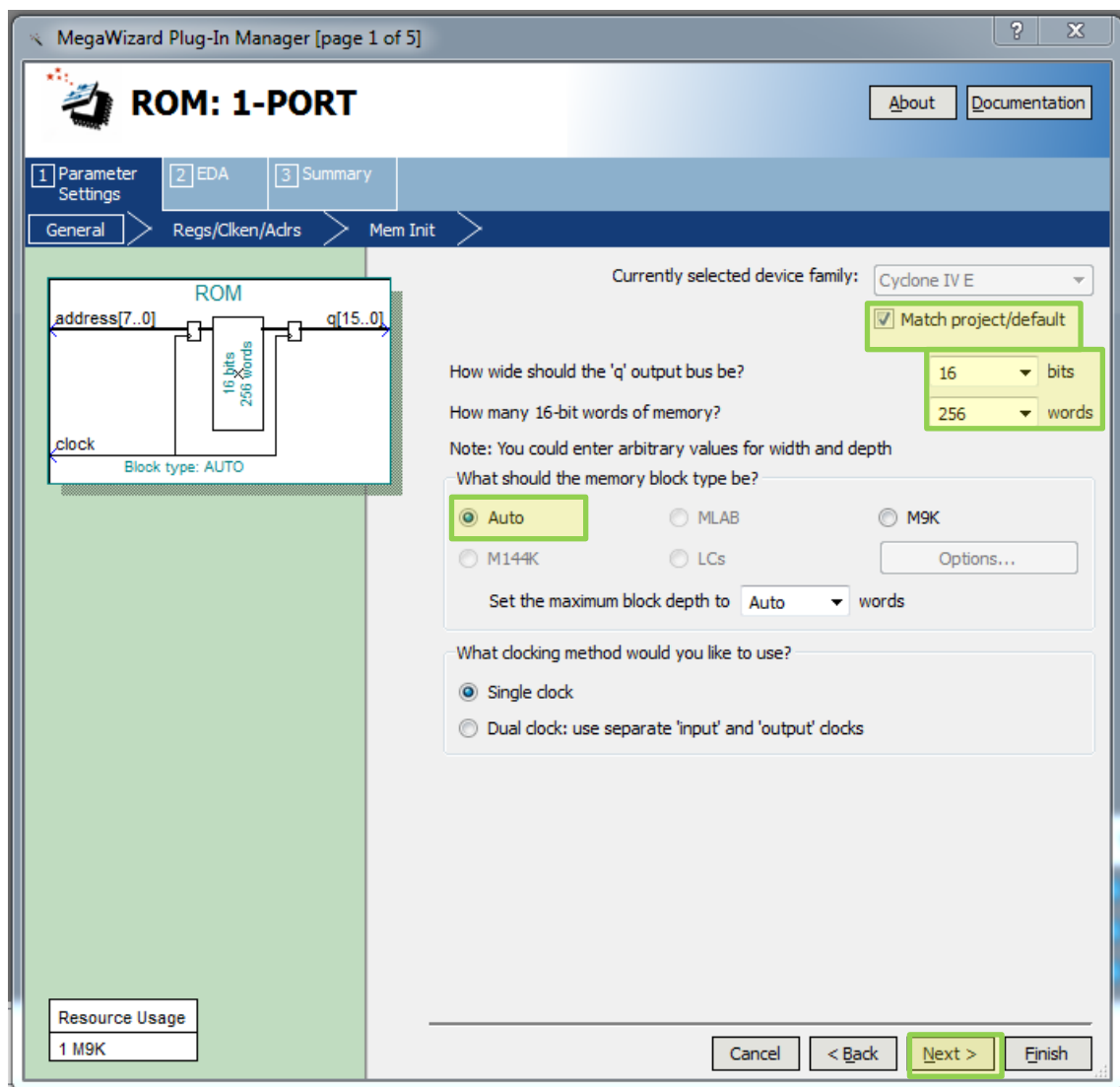


Рис. 11 Окно настройки экземпляра выбранного компонента (page 1)

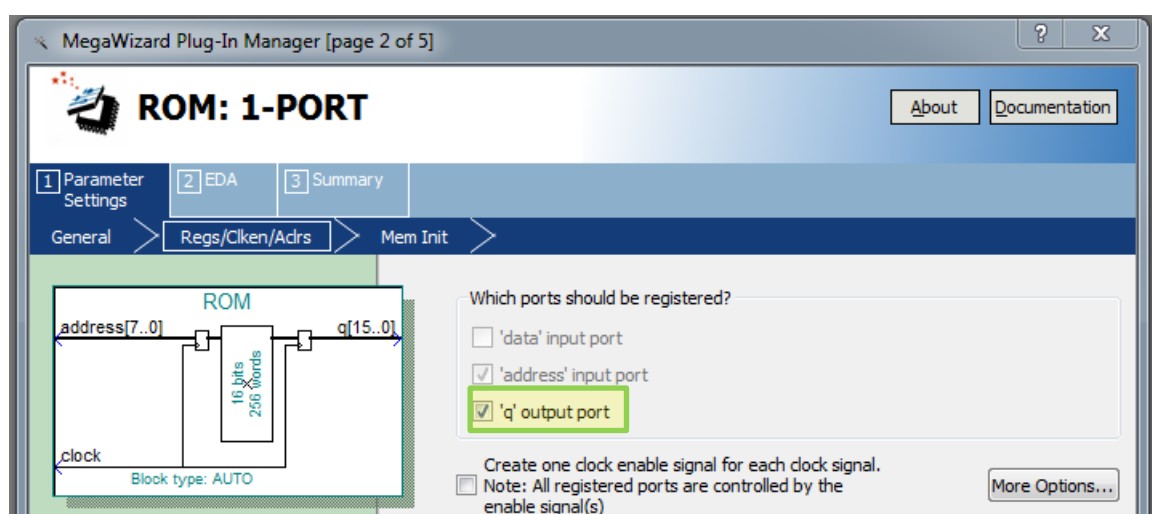


Рис. 12 Окно настройки экземпляра выбранного компонента (page 2)

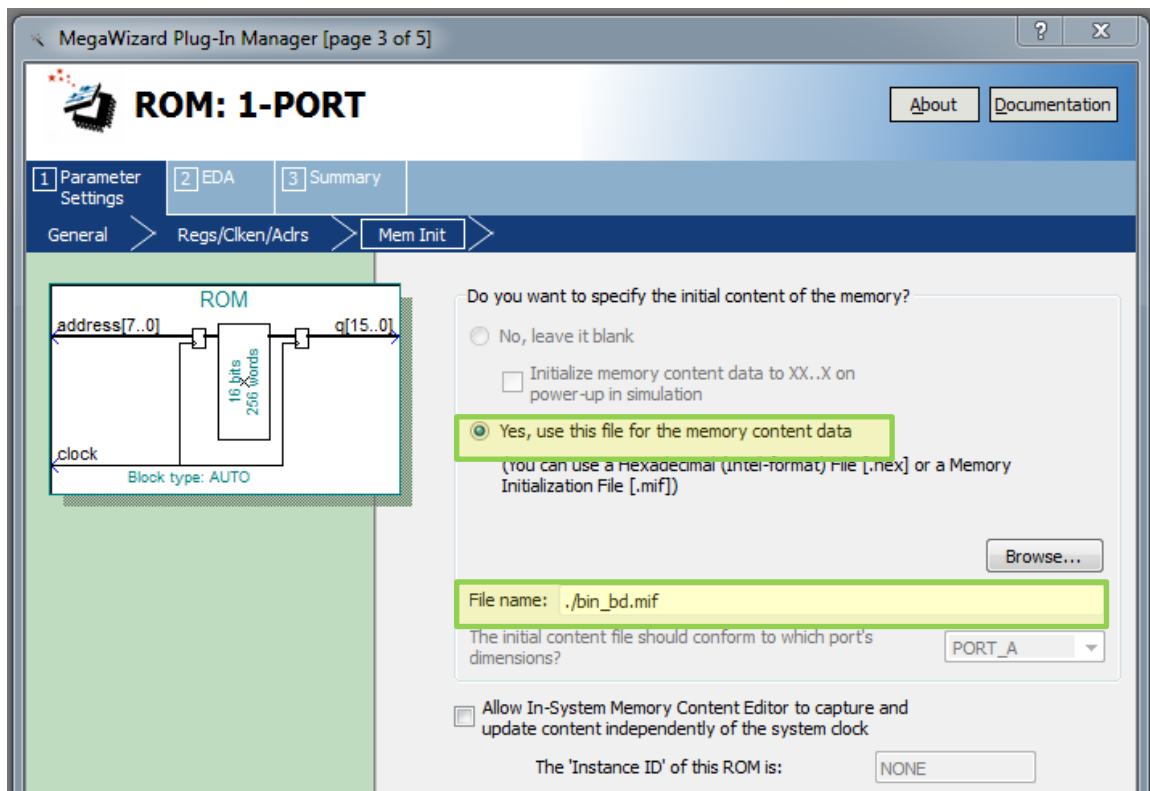


Рис. 13 Окно настройки экземпляра выбранного компонента (page 3)

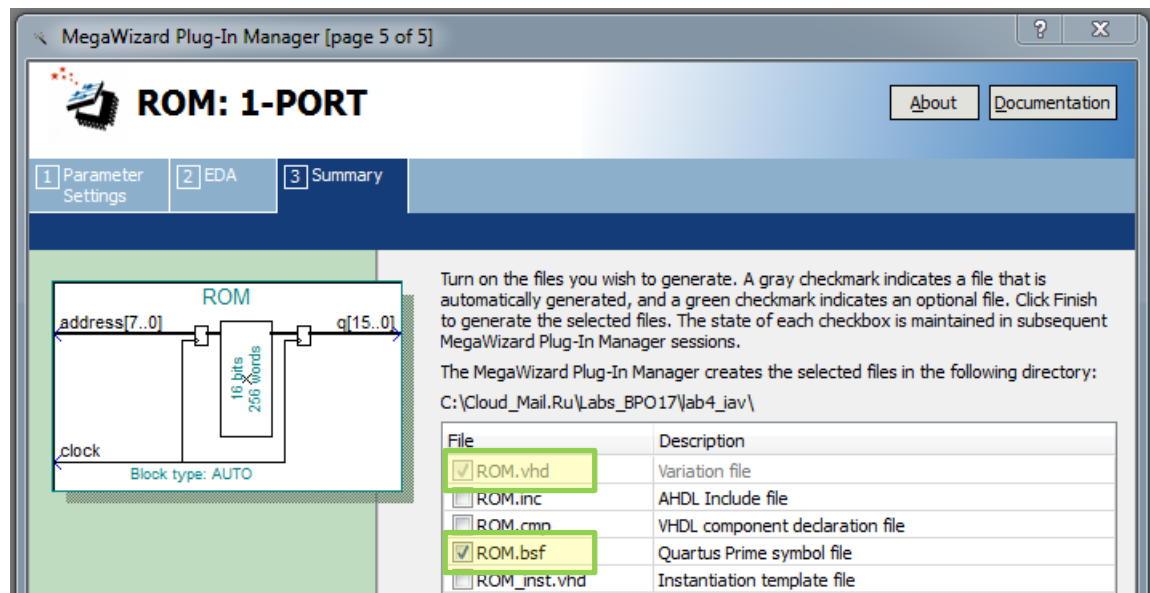


Рис. 14 Окно настройки экземпляра выбранного компонента (page 5)

4.8. В появившемся окне “**QUARTUS Prime IP Files**”, на предложение добавить созданный компонент в проект, выберите «**Yes**». Убедитесь, что файл компонента **ROM.qip** присутствует в окне **Project Navigator** (на вкладке **Files**).

## 5. Создание экземпляра умножителя на базе библиотеки IP-компонентов

5.1. В окне **IP Catalog** в разделе **Library->Basic Functions->Arithmetic** выберите мегафункцию **LPM\_MULT** (рис. 15) и нажмите кнопку «+ADD»,

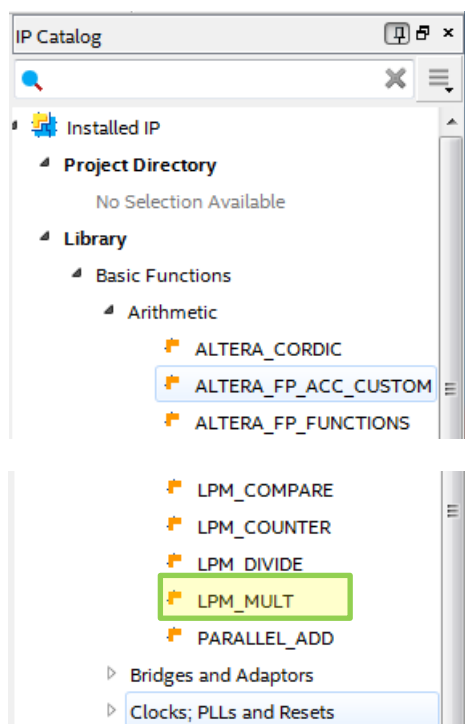


Рис. 15 Окно **IP Catalog**

5.2. В появившемся окне настройки задайте имя экземпляра выбранного компонента (MULT) и укажите язык описания аппаратуры (VHDL), в коде которого он будет храниться в проекте (рис.16). Выполните «OK».

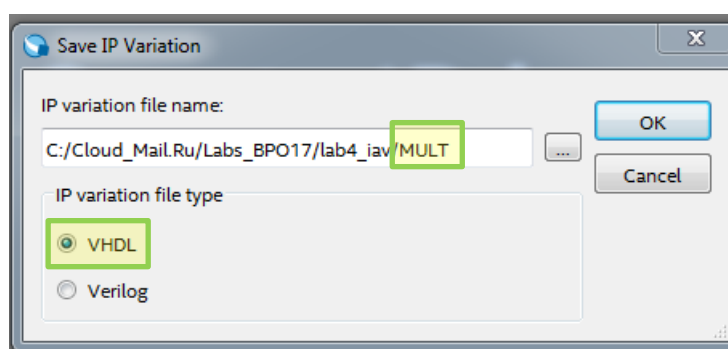


Рис. 16 Окно создания экземпляра выбранного компонента

5.3. В появившемся окне “**MegaWized Plug-In Manager [page 1 of 5]**” установите разрядность **4 бит** и опции как на рис. 17. Выполните «Next».

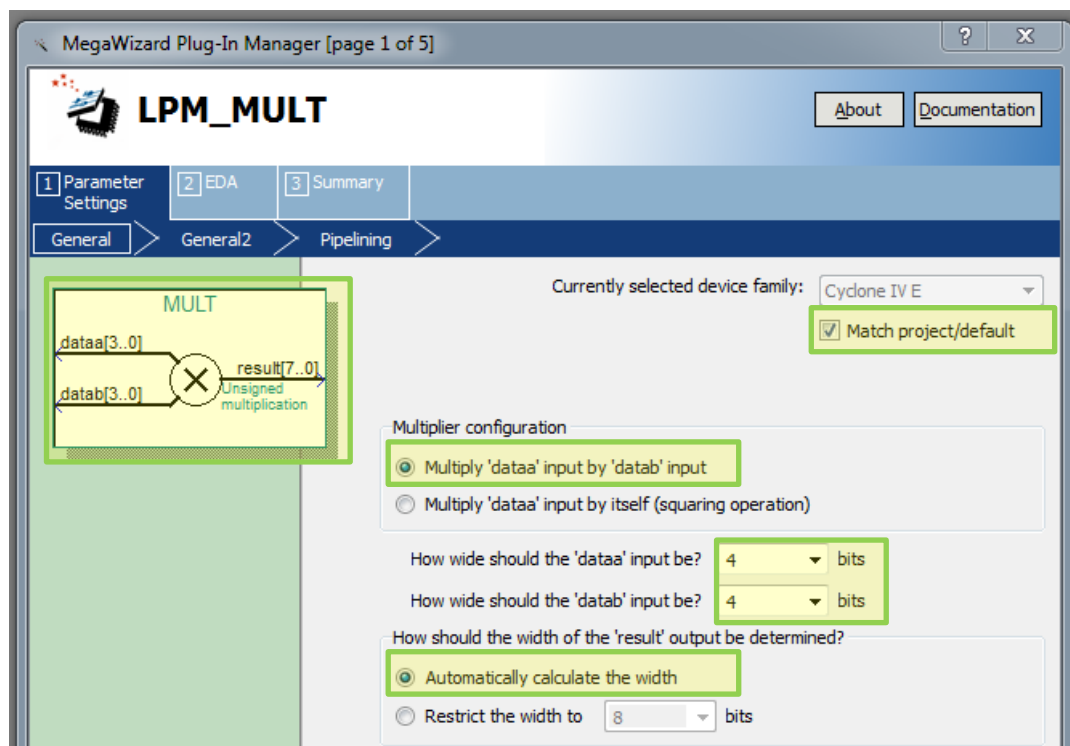


Рис. 17 Окно настройки экземпляра выбранного компонента (page 1)

5.4. В появившемся окне “**MegaWizeded Plug-In Manager [page 2 of 5]**” установите параметры согласно рис. 18.

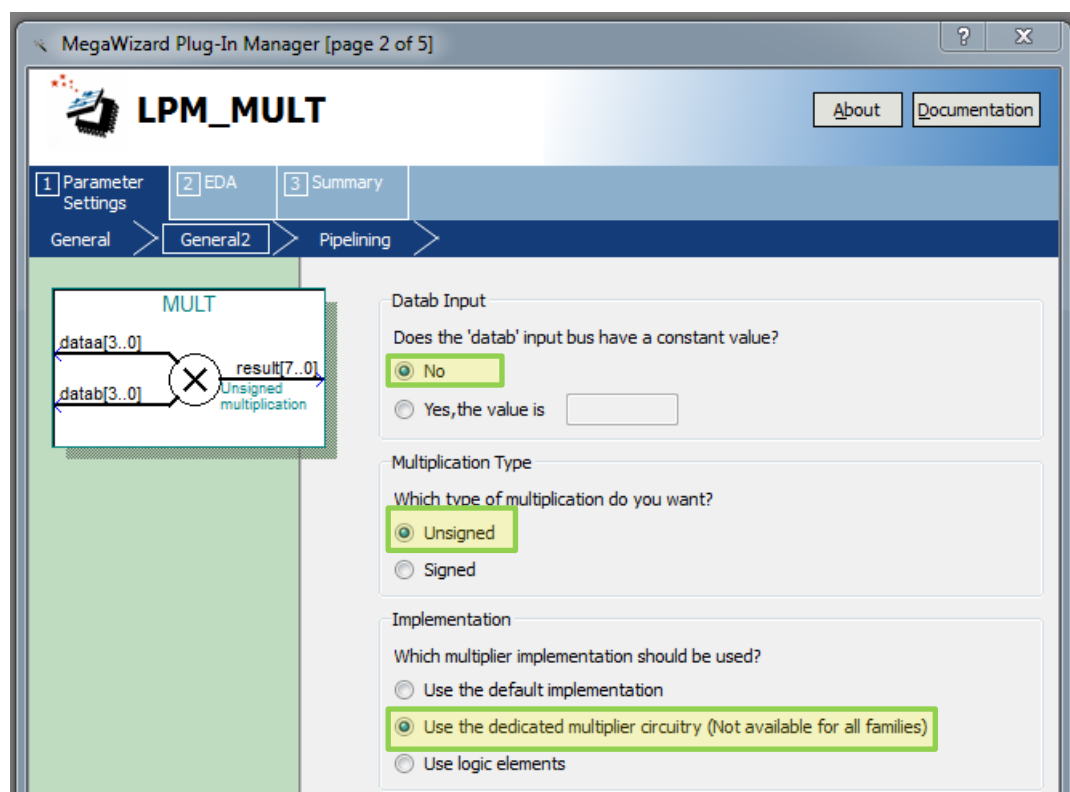


Рис. 18 Окно настройки экземпляра выбранного компонента (page 2)

Выбор опции «**Use the dedicated multiplier**» дает указание компилятору использовать встроенные узлы умножителей в микросхеме ПЛИС. Альтернативно, можно реализовать умножители на логических ячейках (опция «**Use logic elements**»). При выборе опции «**Use the default implementation**» компилятор выберет реализацию, которая указана в настройках компилятора (может настраиваться пользователем). Выполните «**Next**».

5.5. В появившемся окне “**MegaWizared Plug-In Manager [page 3 of 5]**” включите опцию конвейеризации вычислений на 2 такта (рис. 19). Выполните «**Next**».

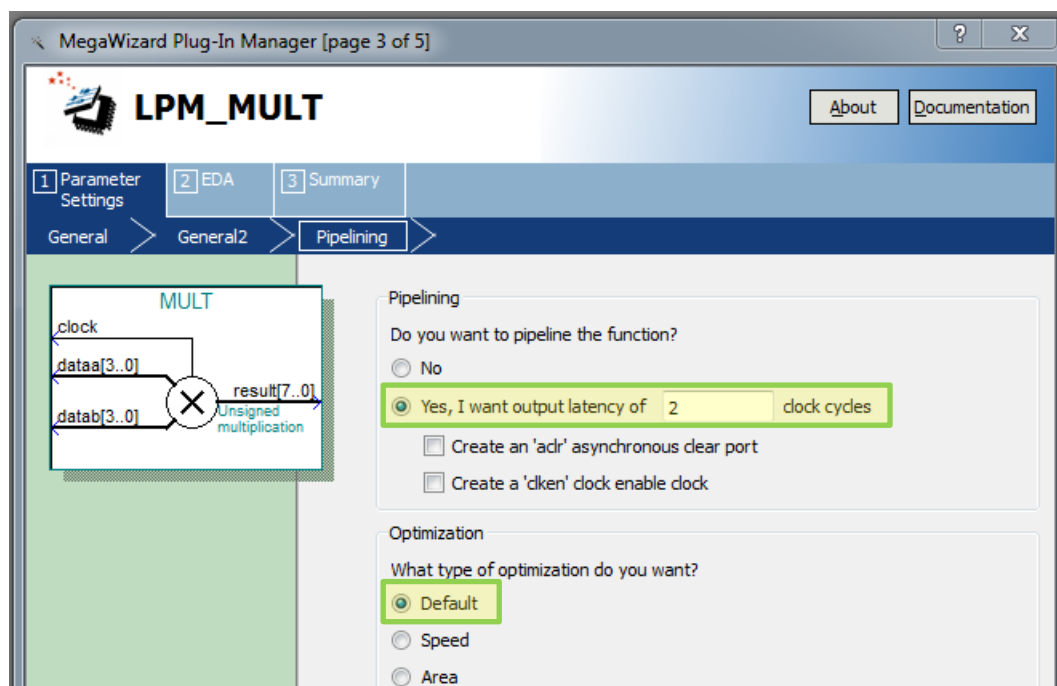


Рис. 19 Окно настройки экземпляра выбранного компонента (page 3)

5.6. В появившемся окне “**MegaWizared Plug-In Manager [page 4 of 5]**” выполните «**Next**».

5.7. В появившемся окне “**MegaWizared Plug-In Manager [page 5 of 5]**” выберите создаваемые файлы, как на рис. 20. Далее «**Finish**». Экземпляр умножителя **MULT** создан.

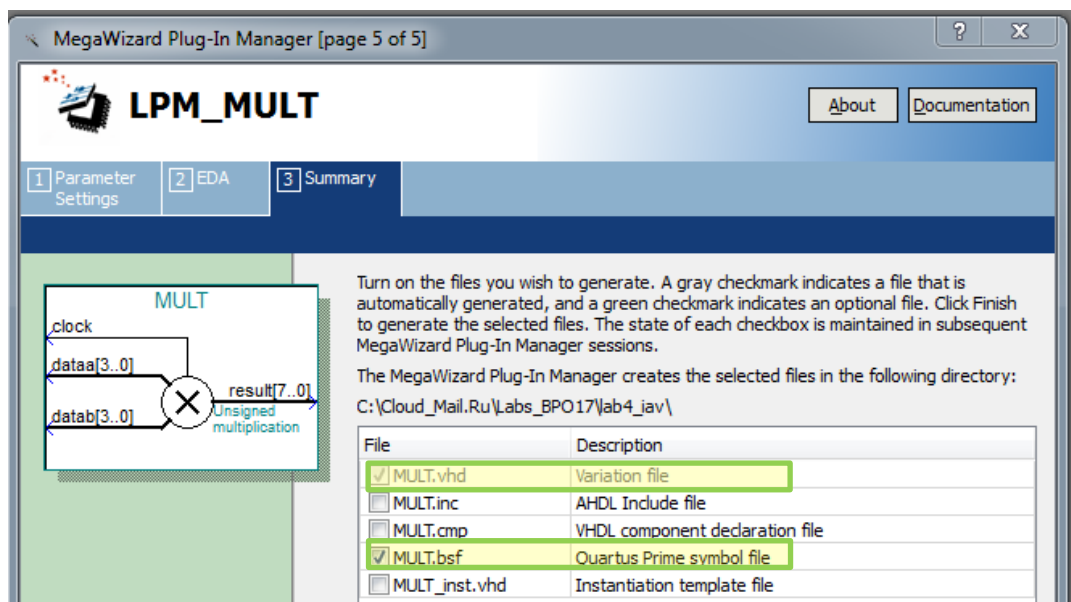


Рис. 20 Окно настройки экземпляра выбранного компонента (page 5)

5.8. В появившемся окне “**QUARTUS Prime IP Files**”, на предложение добавить созданный компонент в проект, выберите «Yes».

## 6. Создание экземпляра умножителя тактовой частоты и блока управления глобальным тактовым сигналом на базе библиотеки IP-компонентов

6.1. В окне **IP Catalog** в разделе

**Library=>Basic Functions=>Clocks; PLLs and Resets=>PLL** выберите мегафункцию **ALTPLL** (рис. 21) и нажмите кнопку «**+ADD**».

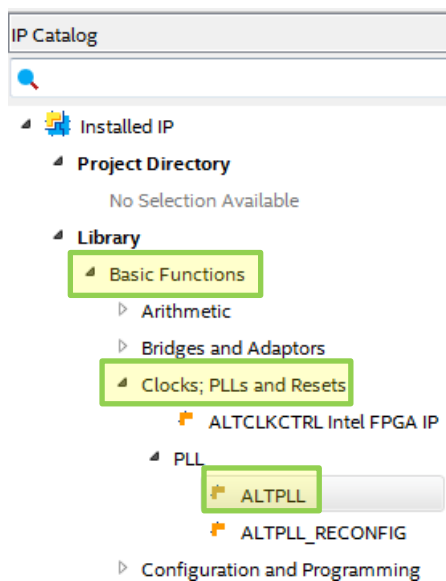


Рис. 21 Окно **IP Catalog**

6.2. В появившемся окне настройки задайте имя экземпляра выбранного компонента (**PLL\_100**) и укажите язык описания аппаратуры (VHDL), в коде которого он будет храниться в проекте (рис.22). Выполните «**OK**».

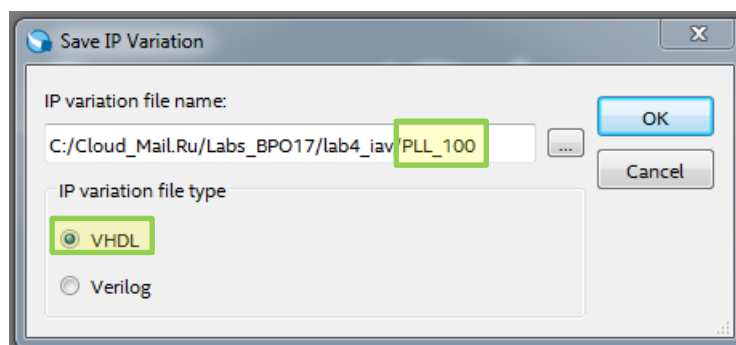


Рис. 22 Окно создания экземпляра выбранного компонента

6.3. В появившемся окне “**MegaWized Plug-In Manager [page 1 of 12]**” установите параметры согласно рис. 23. Выполните «**Next**».

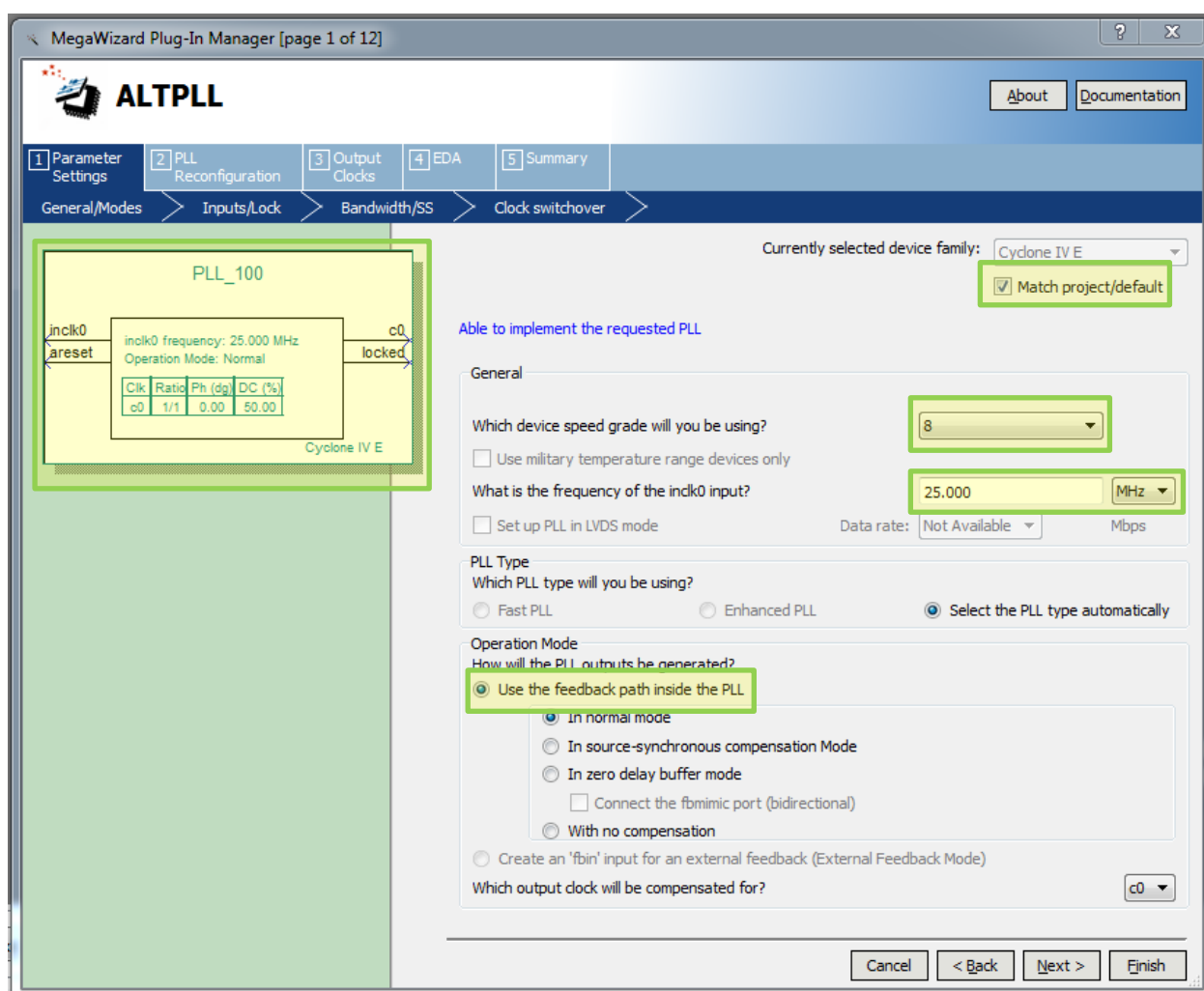


Рис. 23 Окно настройки экземпляра выбранного компонента (page 1)

6.4. В появившемся окне **“MegaWized Plug-In Manager [page 2 of 12]”** установите необходимые опции (рис. 24) и выполните **«Next»**.

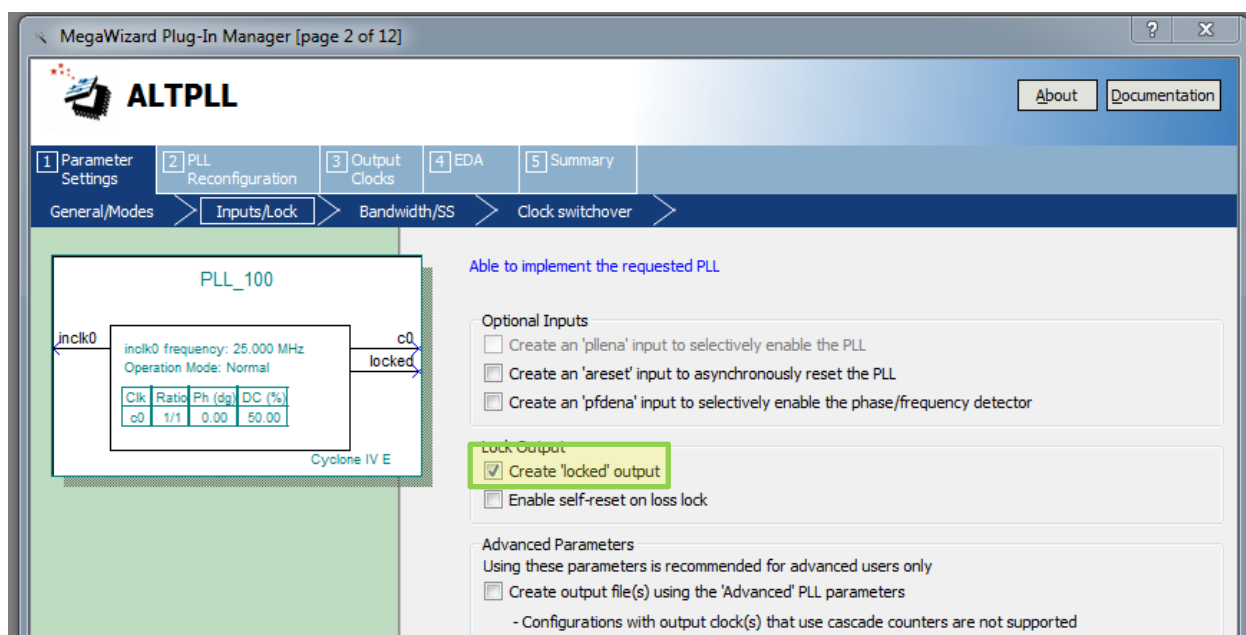


Рис. 24 Окно настройки экземпляра выбранного компонента (page 2)

6.5. На страницах 3-5 **“MegaWized Plug-In Manager ”** выполните **«Next»**.

6.6. В появившемся окне **“MegaWized Plug-In Manager [page 6 of 12]”** выберите опции как на рис. 25.

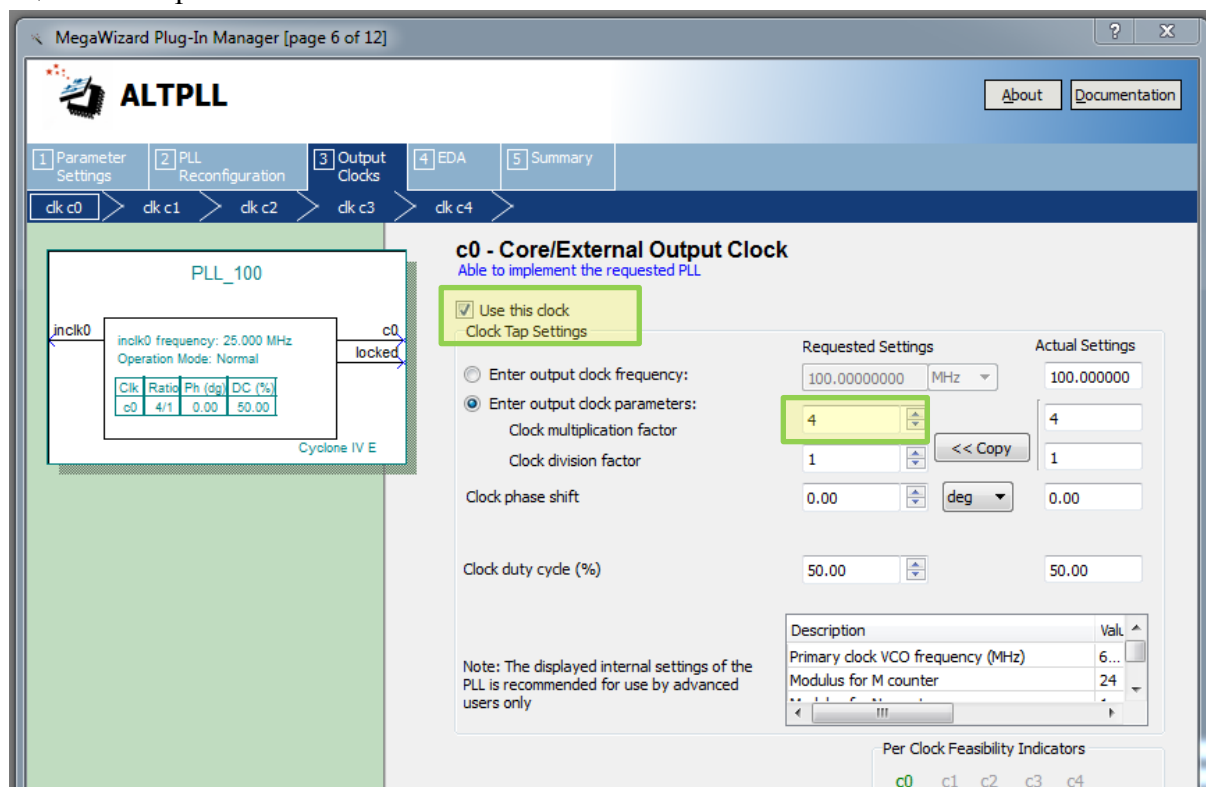


Рис. 25 Окно настройки экземпляра выбранного компонента (page 6)



- 6.7. На страницах **7-11 “MegaWizard Plug-In Manager”** выполните «**Next**».
- 6.8. В появившемся окне **“MegaWizard Plug-In Manager [page 6 of 12]”** выберите создаваемые файлы, как на рис. 25. Далее выполните «**Finish**». Экземпляр умножителя тактовой частоты **PLL\_100** создан.

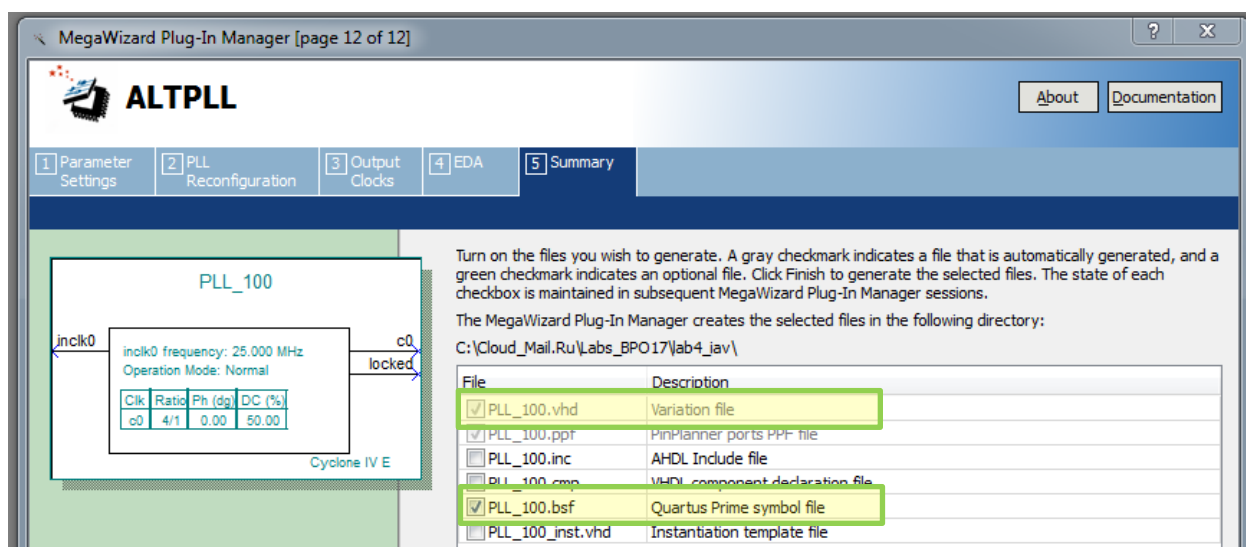


Рис. 25 Окно настройки экземпляра выбранного компонента (page 12)

- 6.9. В появившемся окне **“QUARTUS Prime IP Files”**, на предложение добавить созданный компонент в проект, выберите «**Yes**».
- 6.10. В окне **IP Catalog** в разделе **Library=>Basic Functions=>Clocks; PLLs and Resets** выберите мегафункцию **ALTCLKCTRL\_Intel\_FPGA\_IP** (рис. 26) и нажмите кнопку «**+ADD**».

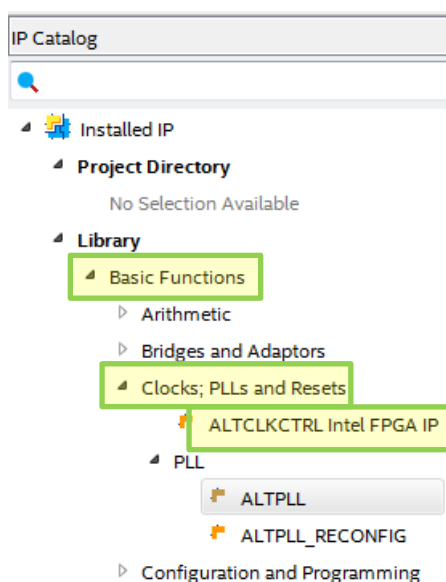


Рис. 26 Окно IP Catalog

6.11. В окне настройки **IP Parameter Editor** появится дополнительное окно **New IP Variation**, задайте в нем в строке **Entity name** имя экземпляра выбранного компонента (**CLK\_CNTR**) (рис.27). Выполните «**OK**».Окно **New IP Variation** закроется.

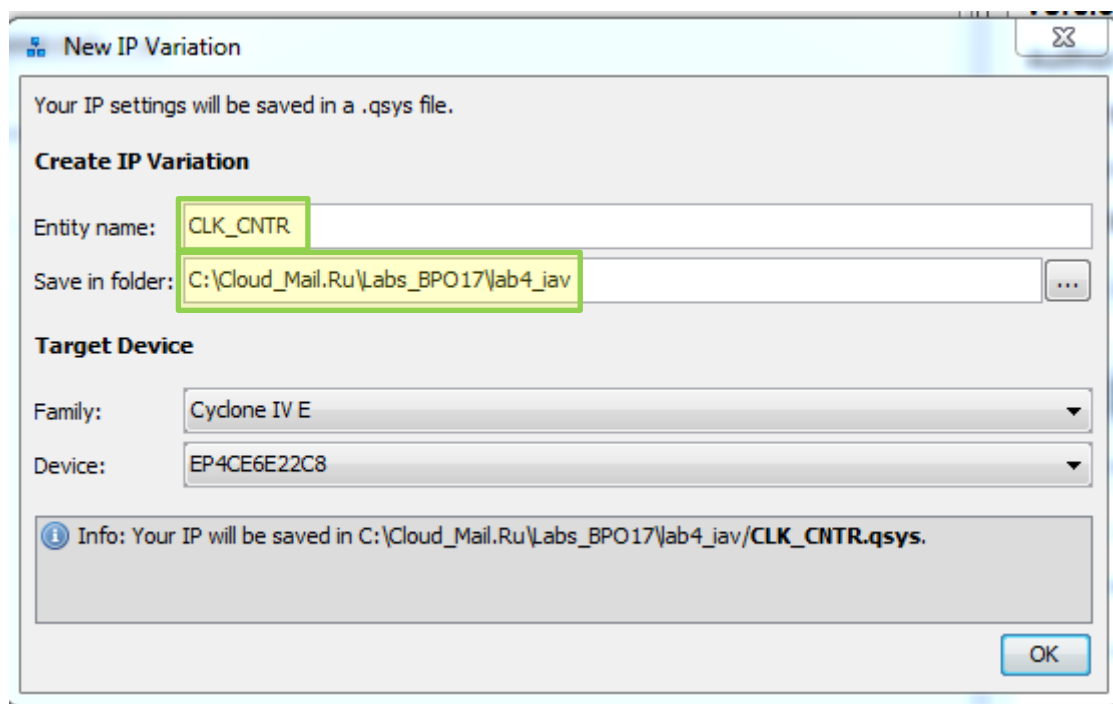


Рис. 27 Окно именования экземпляра выбранного компонента

6.12. В **IP Parameter Editor** установите опции как на рис. 28.

Выполните «**Generate HDL...**».

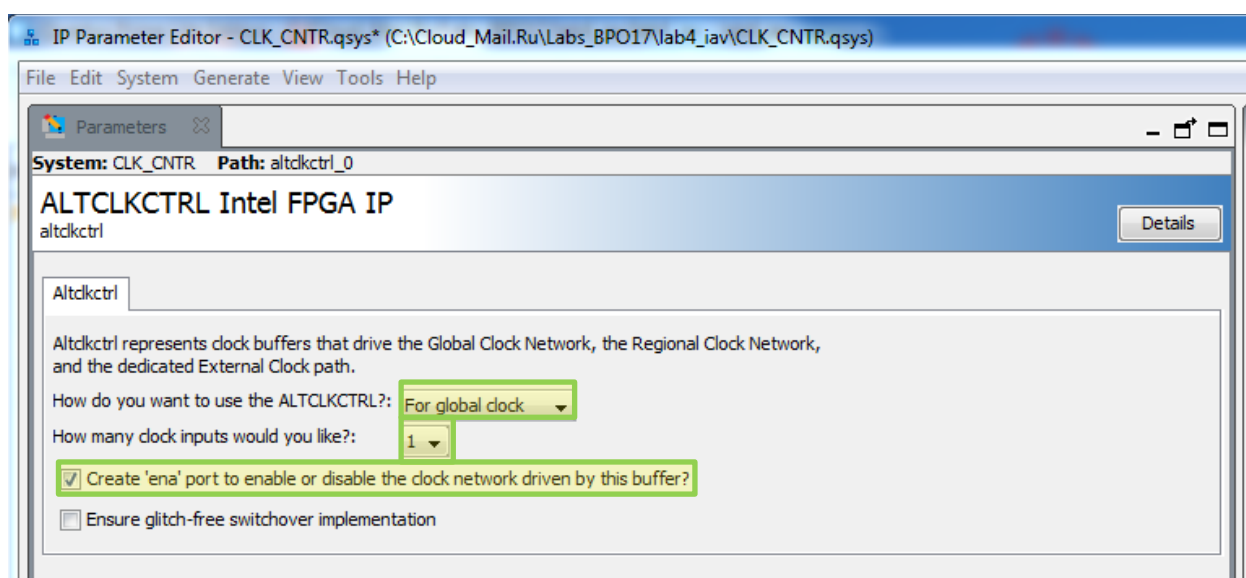


Рис. 28 Окно настройки экземпляра выбранного компонента

6.13. Установите в появившемся окне **Generation** опции как на рис. 29. Выполните «**Generation**».

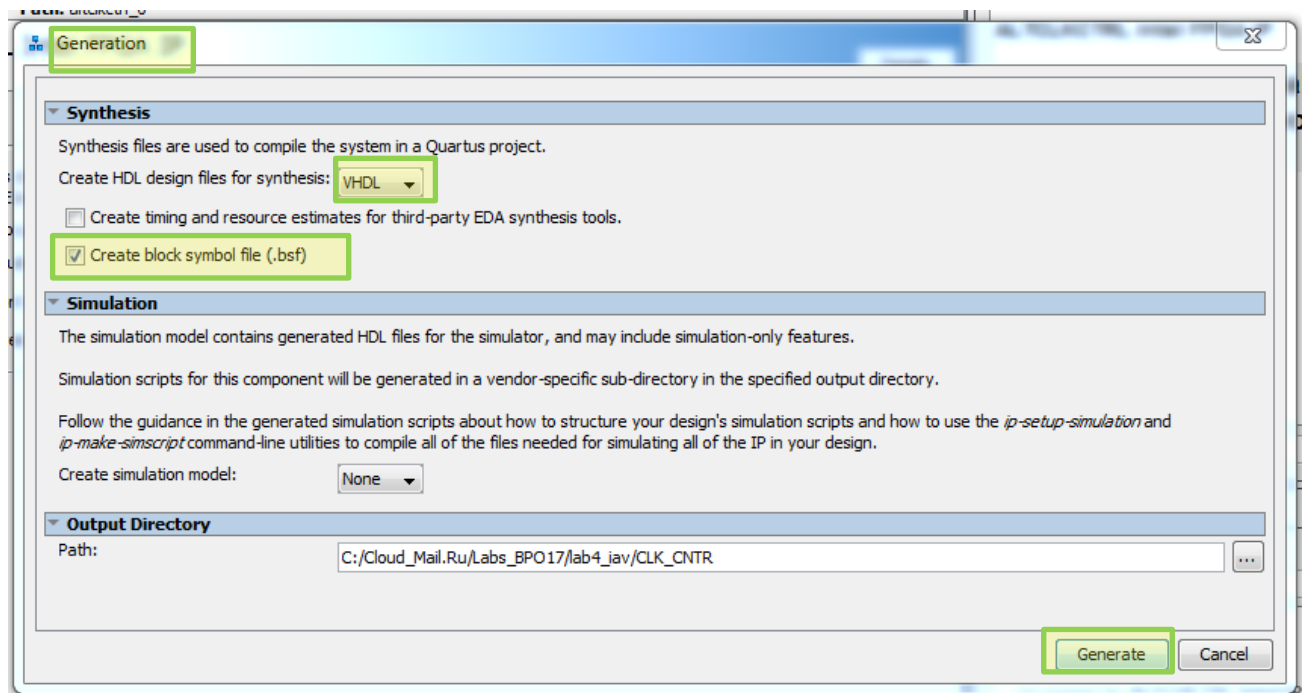


Рис. 29 Окно генерации экземпляра выбранного компонента

6.14. В появившемся окне **Save System Completed** в случае успешного сохранения настроек компонента появится надпись *Save System: completed successfully*. Выполните «Close».

6.15. Появится окно **Generation**, в котором, в случае успешной компиляции, появится надпись *Generation: completed successfully* (рис. 30). Выполните «Close». Затем «Finish».

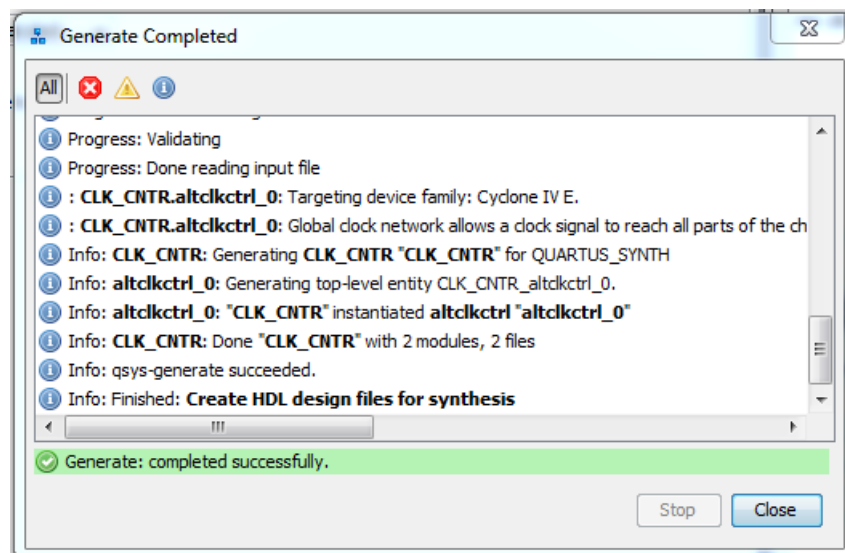


Рис. 30 Окно завершения генерации экземпляра выбранного компонента

6.16. Появится окно **Integration with Quartus Software** с уведомлением о создании файла экземпляра компонента. Выполните «Close».

6.17. Появится окно **Quartus Prime** с уведомлением о необходимости добавить созданные файлы в проект. Там же указаны пути к месту расположения файла. Выполните «Ok».

6.18. Выполните **Project=> Add/Remove Files in Project** и в открывшемся окне **Category**, разделе **Files** в строке **File name** с помощью поиска найдите и выберите файл **CLK\_CNTR.qip**. Выполните «Открыть», затем «Apply» и «Ok». Имя подключенного компонента должно появиться в **Project Navigator**.

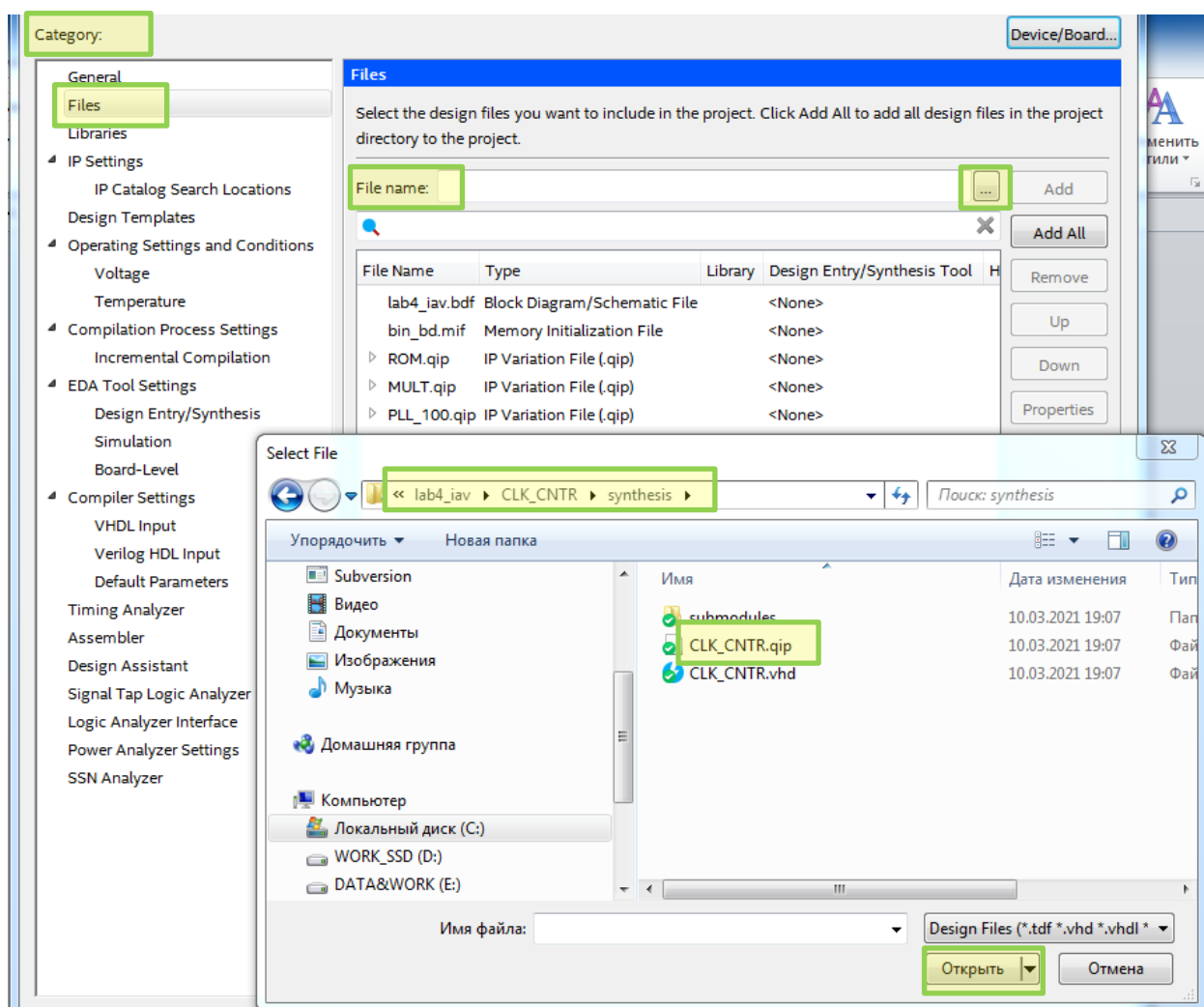


Рис. 31 Подключение к проекту экземпляра созданного компонента

## 7. Создание схемы проекта на базе созданных компонентов и его функциональное моделирование

7.1. Выполните **File => New** или **Create New Design** в окне **Task**, выберите **Block Diagram/Schematic File**, далее «OK». В результате откроется окошко схематического редактора (**Schematic Editor**).

7.2. Нарисуйте схему проекта на базе созданных выше компонентов и компонента **ind\_drv** из проекта **lab3\_iav**, подключенного в качестве внешней библиотеки (рис. 32).

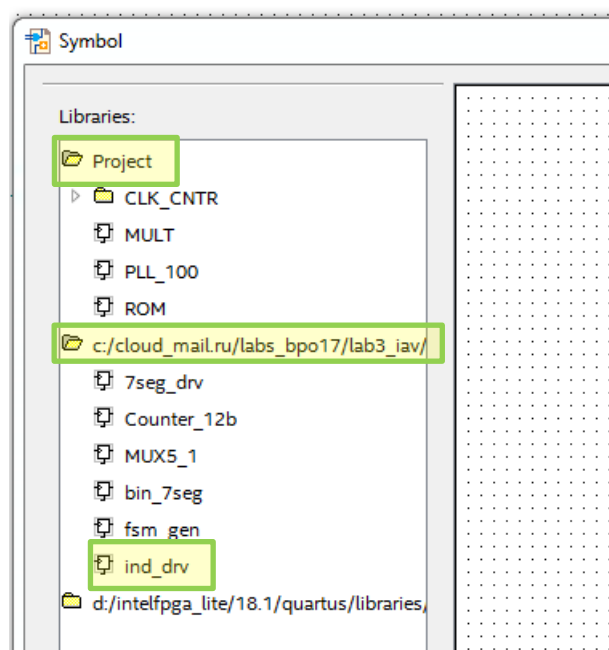


Рис. 32 Компоненты текущего проекта и компоненты проекта, подключенного в качестве внешней библиотеки

7.3. Сохраните файл со схемой **File -> Save As** с именем **lab4\_iav.bdf** (рис. 33) .

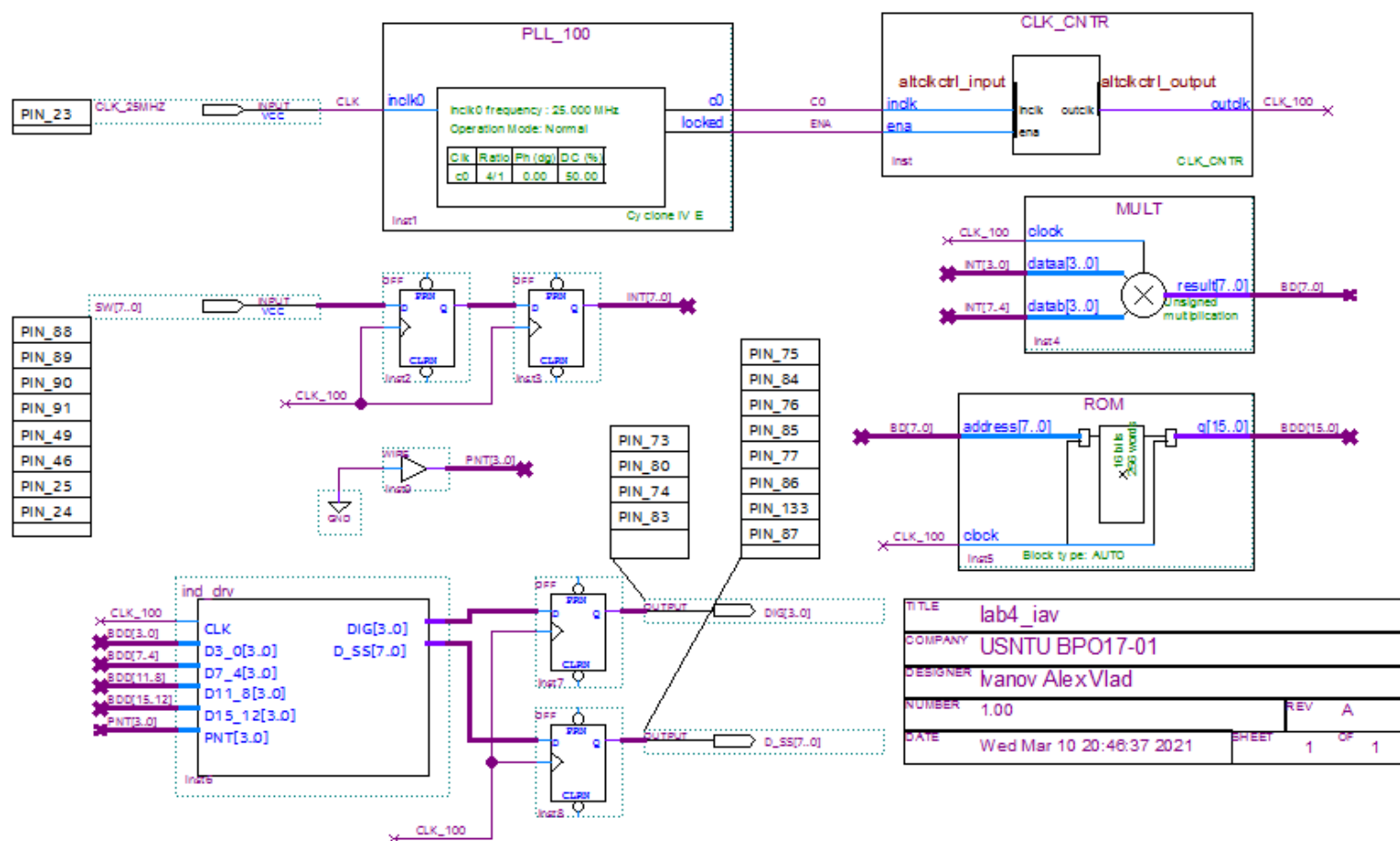


Рис. 33 Схема дизайн-файла проекта **lab4\_iav**

Номера выводов появятся только после назначения номеров выводов микросхемы, к которым подключаются сигналы схемы (см. п.10.2.). Видимость номеров микросхемы на схеме управляется следующим образом: кликните ПКМ на поле редактора схемы, выберите **Show => Show Location Assignments**.

7.4. Дважды кликните левой клавишей мыши на компонент **ind\_drv**, затем в открывшейся схеме выберите счетчик **Counter\_12b**, кликните ПКМ и в выпавшем меню выберите опцию **IP Parameter Editor**. Откроется редактор компонента **Mega Wizard Plug-In Manager**, на стр.2 установите модуль счетчика равным **4**. Далее выполните «**Finish**» два раза. В открывшемся окне с предложением выполнить **update** символа **Counter\_12b** выполните «**Yes**». Далее в окне **Update Symbol or block** выполните «**Ok**». Убедитесь, что на изображении символа **Counter\_12b** модуль счета равен **4**. Сохраните файл **ind\_drv** и закройте его.

7.5. Сохраните файл **lab4\_iav**.

## 8. Функциональное моделирование проекта

8.1. Установите файл **lab4\_iav.bdf** проектом верхнего уровня:

**Project => Set as Top-Level Entity.**

8.2. Выполните компиляцию **Analysis@Synthesis**. Проконтролируйте имя проекта верхнего уровня в отчете компилятора (Окно **Compilation Report**). Устраните ошибки при необходимости.

8.3. Выполните **File => New**, откроется окошко с выбором типов создаваемых файлов. Выберите **University Program VWF**, далее «**OK**». В результате откроется окошко редактора эпюр симуляции (**Simulation Waveform Editor**).

8.4. Сохраните созданный файл под именем **lab4\_iav.vwf (File => Save\_As)**. В окошке редактора сделайте следующие настройки:

- шаг сетки на экране            **10 ns:**            **Edit -> Grid Size;**

- время симуляции            **2000 ns:**            **Edit -> Set End Time;**

8.5. Задайте в колонке **Name** редактора эпюр имена выводов и сигналов проекта. Для этого выполните: **Edit => Insert-> Insert Node or Bus** (или в поле колонки «**Name**», кликните ПКМ и выберите из выпадающего списка «**Insert Node or Bus**»). В появившемся окошке выберите «**Node Finder...**». Далее в появившемся окошке **Node Finder** (рис. 34) в окошке **Filter** выберите «**Pins: all**», нажмите «**List**».

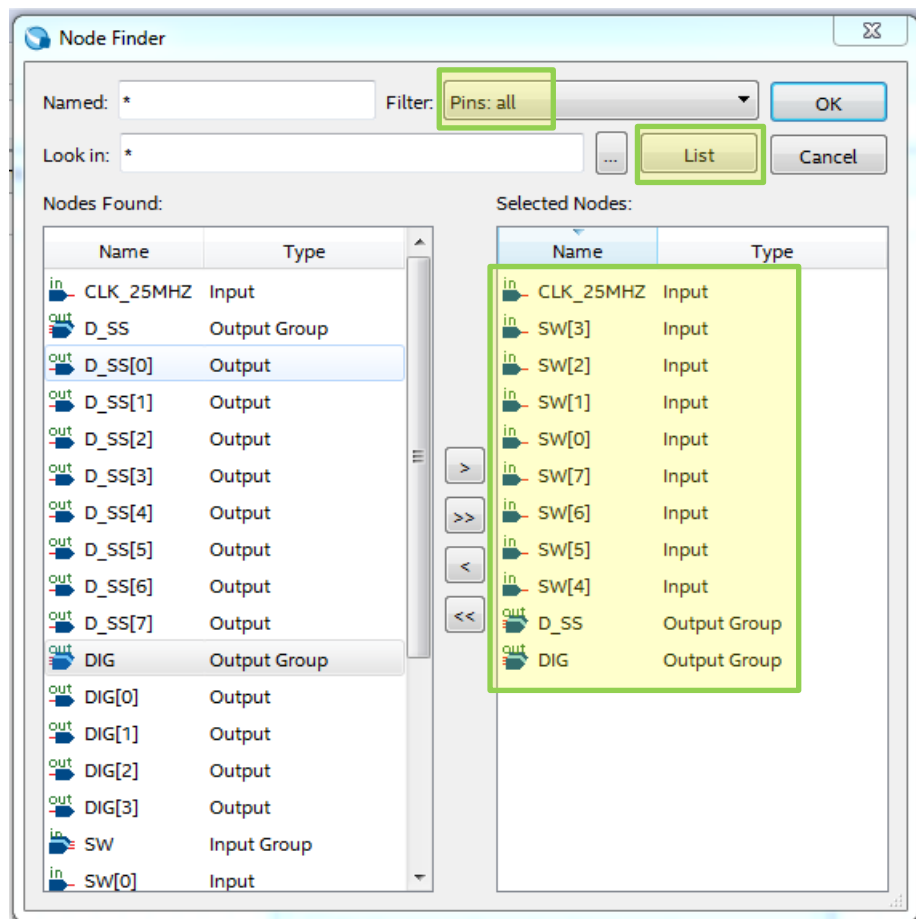





Рис.34 Выбор сигналов для симуляции

В окошке **Nodes Found** выделите необходимые сигналы и перенесите в окошко **Selected Nodes** с помощью кнопки  (по одному сигналу или группу выделенных сигналов, при этом можно сразу упорядочить сигналы, чтобы было удобнее наблюдать их при анализе результатов симуляции) или кнопку  (все сигналы). Попробуйте сразу упорядочить сигналы, это позволит при анализе эпюр сигналов лучше понять работу схемы. Многоразрядные выходные сигналы выберите как групповые. Входные выберите поразрядно. Далее дважды выполните «Ok».

8.6. Сгруппируйте сигналы **SW3..SW0** в группу **SW\_A**, сигналы **SW7..SW4** в группу **SW\_B**, систему счисления сигналов для отображения задайте, при создании группы, как **Unsigned Decimal**. Группа создается следующим образом: выделите **Ctrl + ЛКМ** все нужные сигналы группы, затем нажмите **ПКМ**, выберите **Grouping-> Groupe** и в появившемся окне введите имя группы, далее «Ok». После создания группы можно закрыть ее, чтобы отображался только общий сигнал группы (рис. 37).

8.7. Задайте сигнал **CLK\_25MHZ** как периодический с периодом 40ns ()

8.8. Задайте значения сигналов группы **SW\_A** как **Count**. Для этого выделите поле сигнала группы двойным кликом **ЛКМ**, затем **ПКМ=> Value => Count Value** или выберите  из панели инструментов сигналов. В открывшемся окне задайте параметры (рис. 35)

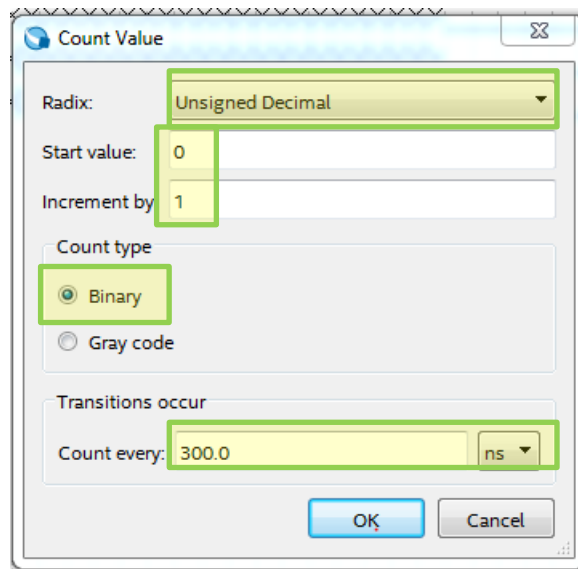



Рис.35 Настройка сигналов группы SW\_A

8.9. Задайте значения сигналов группы **SW\_B** как константу с значением **8**. Для этого выделите поле сигнала группы двойным кликом **ЛКМ**, затем **ПКМ**=> **Value** => **Count Arbitrary** или выберите  из панели инструментов сигналов. В открывшемся окне задайте параметры (рис. 36)

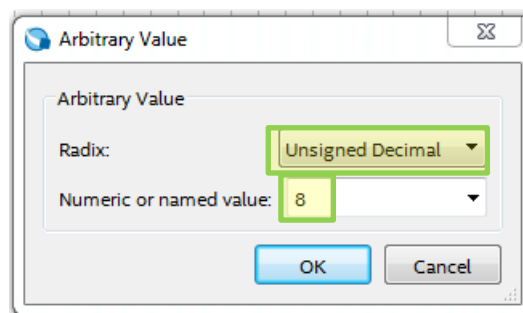


Рис.36 Настройка сигналов группы SW\_B

8.10. Окончательный вид теста показан на рис. 37 . Сохраните созданный тест.



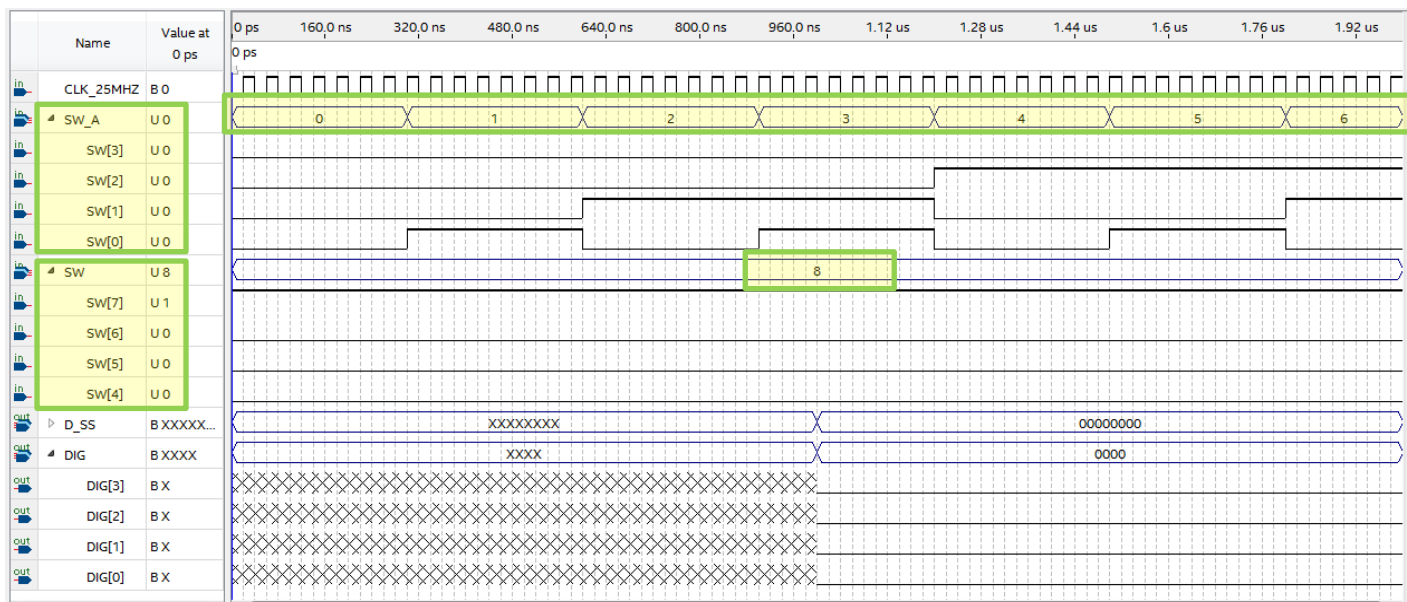



Рис.37 Окончательный вид теста для симуляции

8.11. Выполните в окне редактора эппюр

**Simulation -> Simulation Settings -> Restore Defaults-> Save.**

8.12. Запустите симуляцию нажатием на кнопку  (**Run Functional Simulation**) или выполните **Simulation -> Run Functional Simulation**.

На предложение во всплывшем окошке сохранить изменения в файле симуляции нажмите “Ok”. Запустится функциональная симуляция, появится окно **Simulation Flow Progress** с отображением выполнения команд текущего процесса. В случае успешной симуляции окно **Simulation Flow Progress** закроется и откроется новое окно редактора эппюр (Read-Only) с результатами симуляции (рис. 38). В этом окне можно только просмотреть результаты симуляции, для модификации эппюр надо вернуться в первое окно симулятора, где они создавались. Сохраните скриншот для отчета.

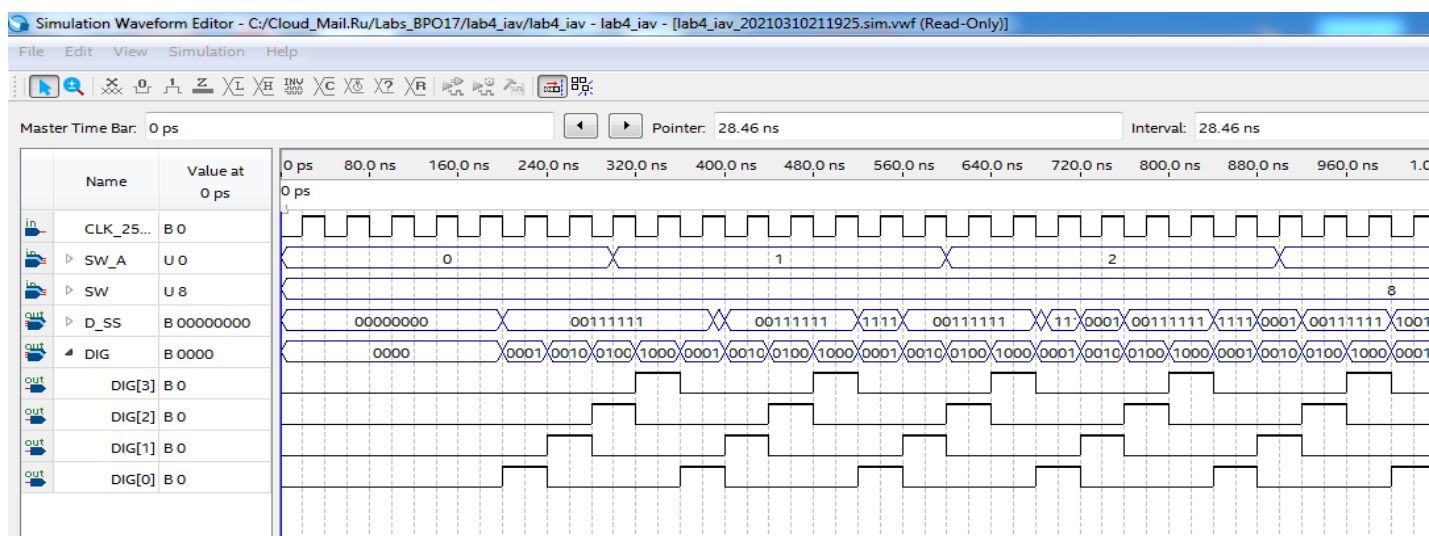


Рис.38 Результат функциональной симуляции схемы проекта

Если окно **Simulation Flow Progress** после симуляции не исчезнет и в нем появятся сообщения об ошибках, выполните следующую процедуру:

- закройте окно **Simulation Flow Progress**
- закройте окно редактора эюр **Simulation Waveform Editor**
- выполните еще раз задачу **Analysis@Synthesis** (проконтролируйте, чтобы проектом верхнего уровня был файл **ind\_drv.bdf**). При этом обновятся все изменения в проекте, связанные с симуляцией.
- после успешной компиляции откройте файл **ind\_drv.vwf** и в открывшемся окне **Simulation Waveform** выполните **Simulation -> Run Functional Simulation**
- убедитесь в правильности результатов симуляции

8.13. Раскройте участок эюр, где происходит умножение числа 3 на 8 и оцените по состоянию сигнала D\_SS правильность работы схемы (рис. 39). Оценивайте два младших разряда индикатора (1 и 2, справа налево), так как числа маленькие и поэтому в 2-х старших разрядах индикатора (3 и 4) изменений нет.

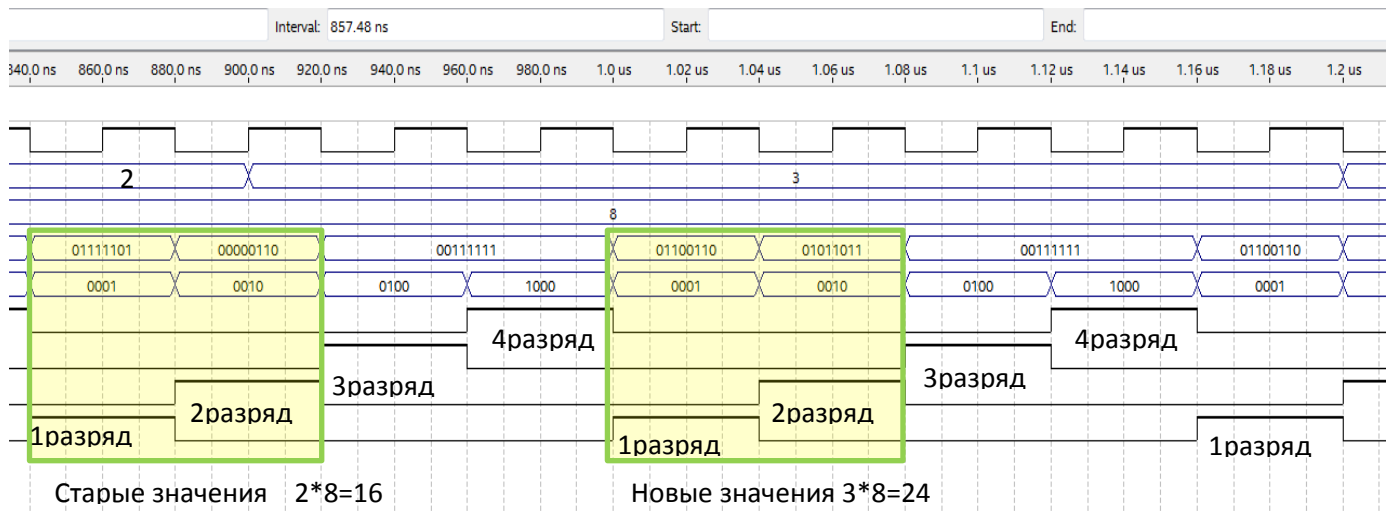


Рис.39 Результат функциональной симуляции схемы проекта (в увеличенном масштабе)

## 9. Анализ проекта с помощью RTL-Viewer, Technology Map Viewer (Post-Mapping)

9.1. Выполните **Tools=> Netlist Viewers=>RTL Viewer**. Откроется окно **RTL viewer**, где можно увидеть схематическое представление проекта в условных графических обозначениях и проанализировать ее функциональное соответствие задаче (рис. 40).

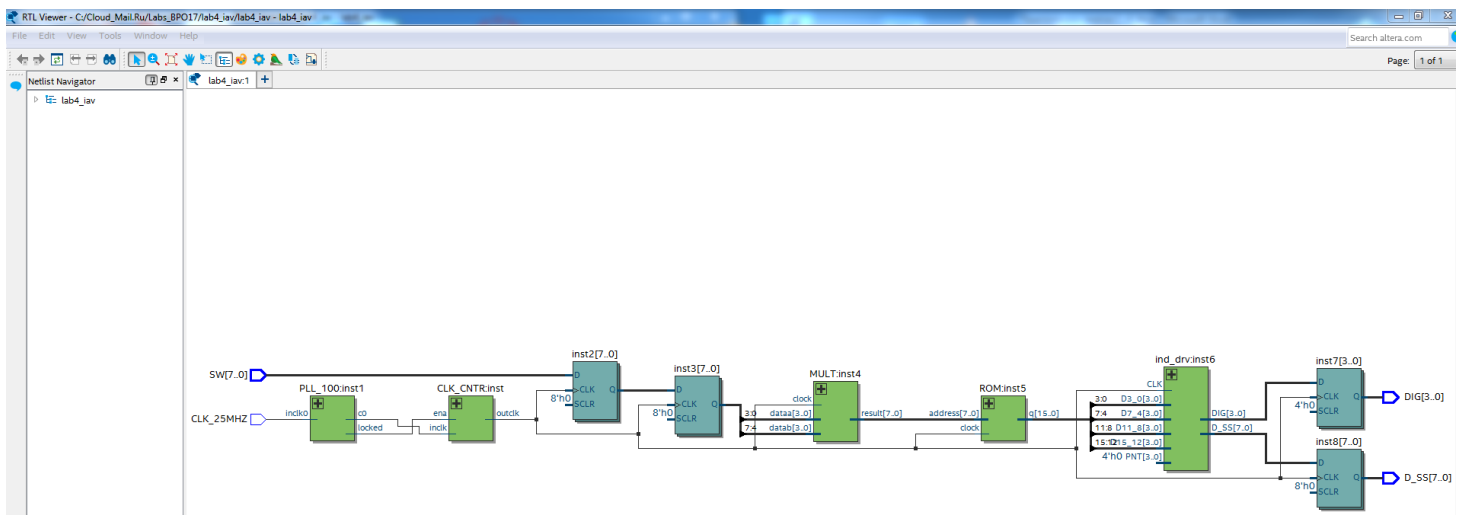


Рис. 40 Представление проектного файла в RTL viewer

9.2. В окне **Netlist Navigator** раскройте строку с названием проекта верхнего уровня и просмотрите всю иерархию проекта (рис. 41) .

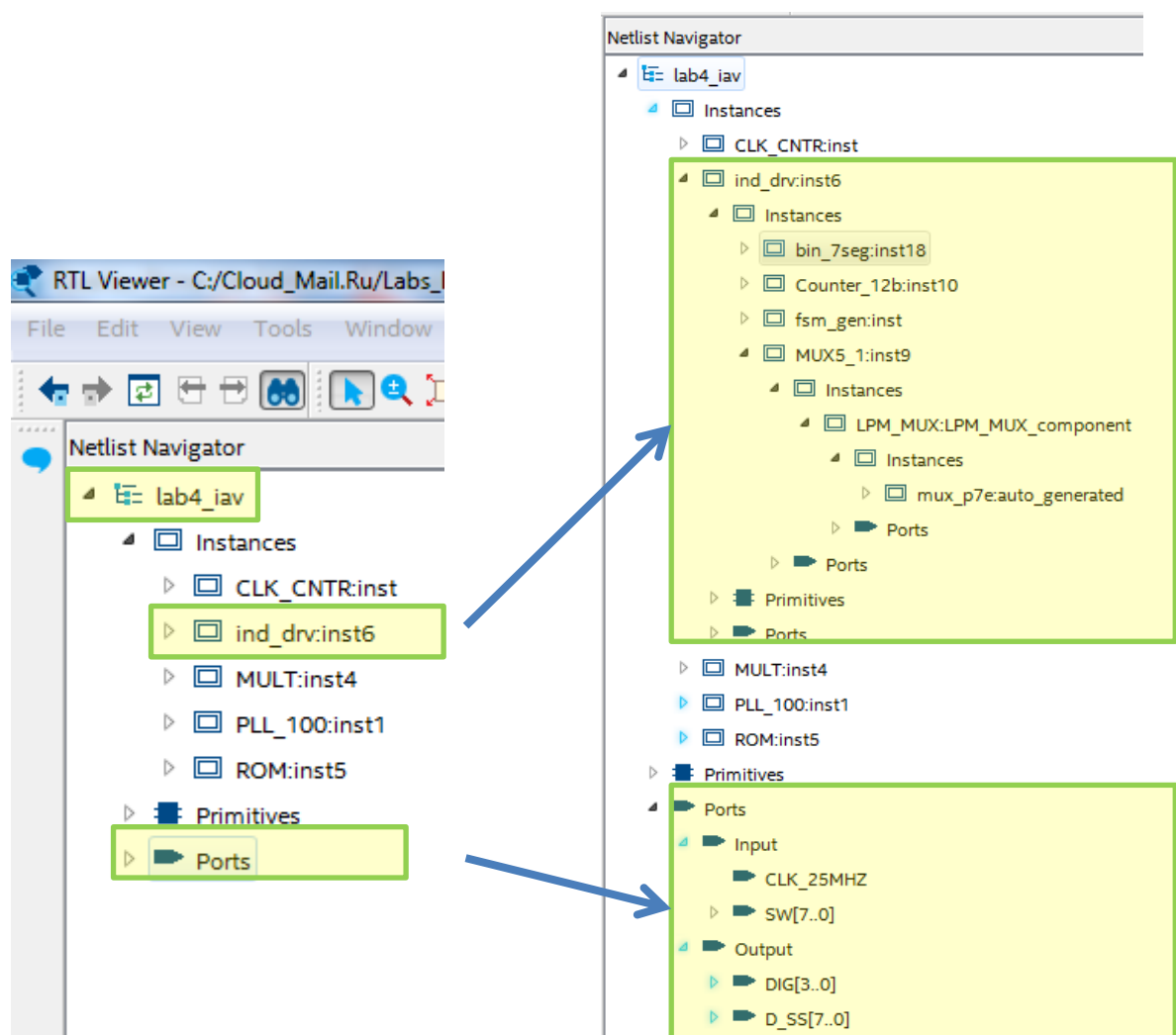


Рис. 41 Окно Netlist Navigator с отображением структуры проекта на разных уровнях (RTL viewer)

9.3. Управляя кнопкой «+» на символах компонентов можно посмотреть их внутренне содержание (рис. 42). Таким образом, можно просмотреть любой элемент схемы до самого нижнего уровня иерархии. Посмотрите, например, внутреннюю структуру символа **bin\_7seg** в компоненте **ind\_drv** до самого нижнего уровня реализации (рис.43).

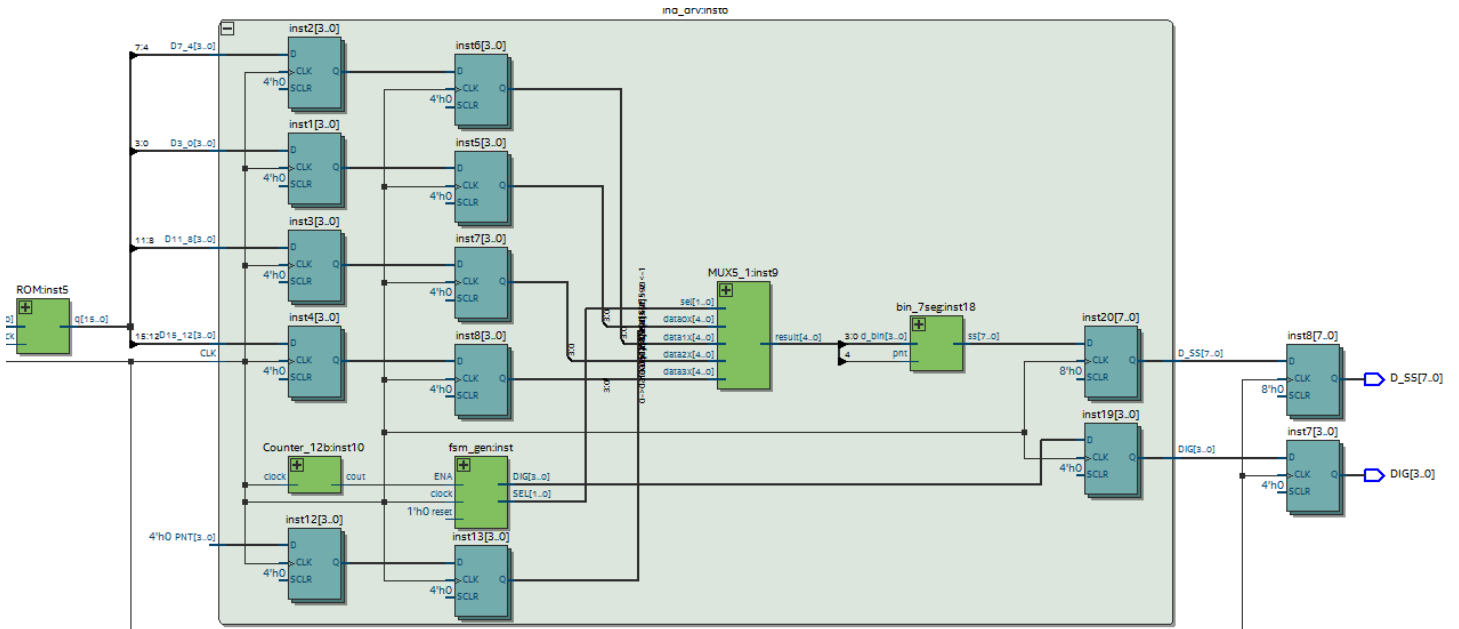


Рис. 42 Просмотр внутренней структуры компонента `ind_drv` в окне RTL viewer

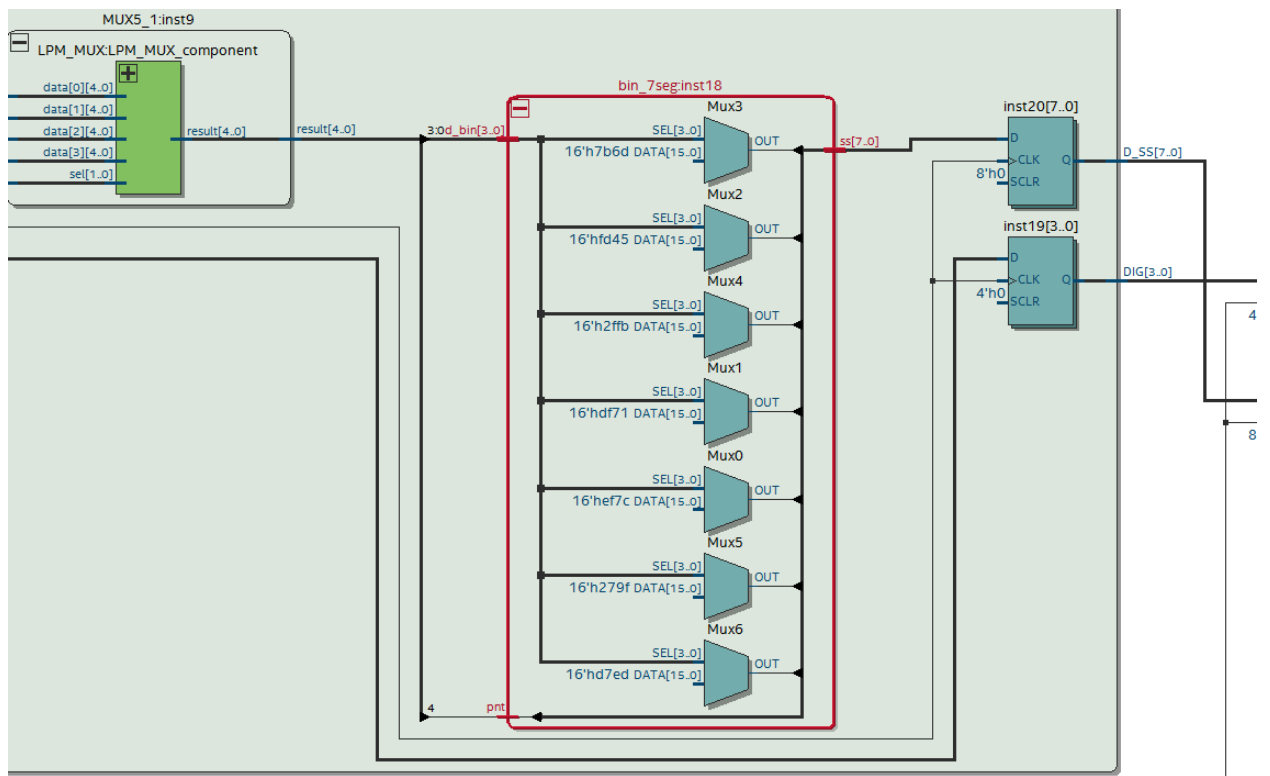


Рис. 43 Реализация компонента `bin_7seg` на элементарном уровне в окне RTL viewer

9.4. Если выделить ЛКМ ввод или вывод компонента, нажать ПКМ и выбрать в меню **Connectivity Details** (рис. 44) , то можно увидеть детальную информацию о сигналах, поступающих на этот вывод (рис. 45).

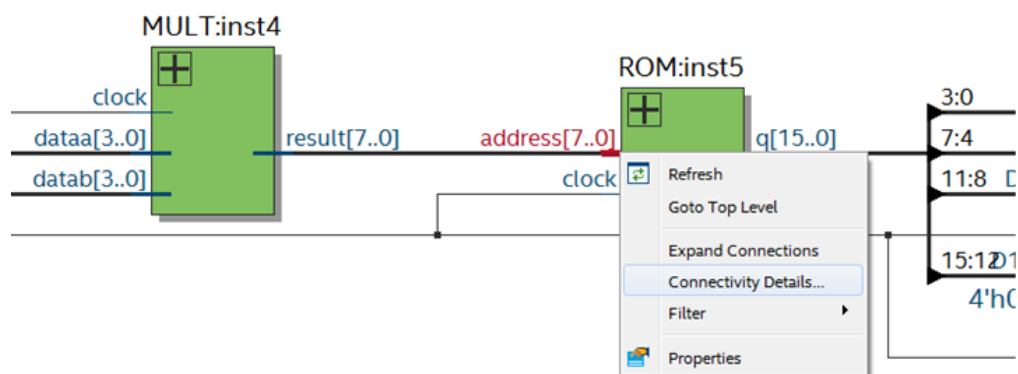


Рис. 44 Просмотр детальной информации о сигналах, поступающих на вывод компонента

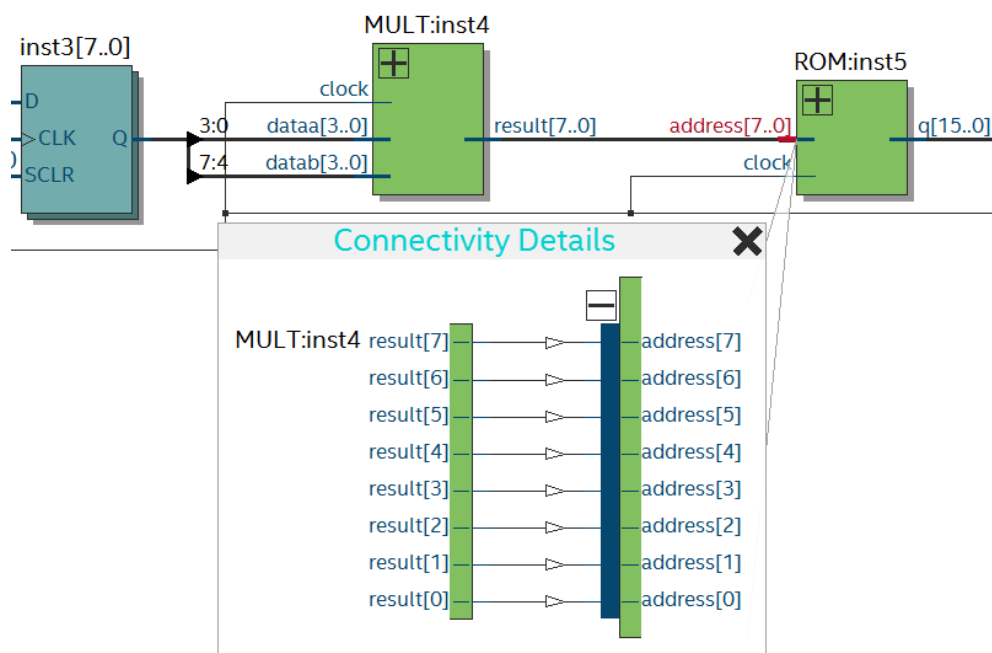


Рис. 45 Детальная информация о сигналах на выводе address[7..0] компонента ROM

9.5. Любой из компонентов проекта можно открыть в новом окне и рассмотреть более детально. Для этого выделите ЛКМ нужный компонент в окне **Netlist Navigator**, нажмите ПКМ и выберите **Display in New Tab**. В новой вкладке отобразится только выбранный компонент проекта (рис. 46)

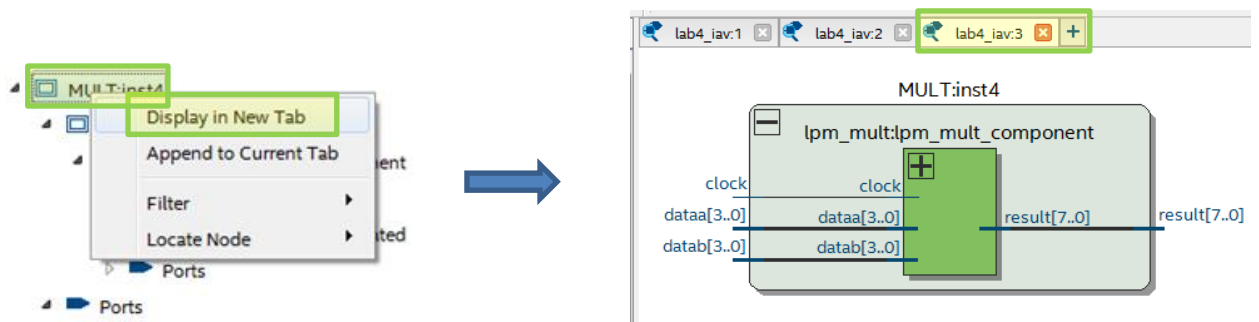


Рис. 46 Просмотр компонента MULT в новом окне

9.6. Выберите ПКМ компонент **PLL\_100** и далее **Locate Node=> Locate in Design File** (рис. 47). Откроется окно схематического редактора с выбранным компонентом в фокусе (рис. 48).

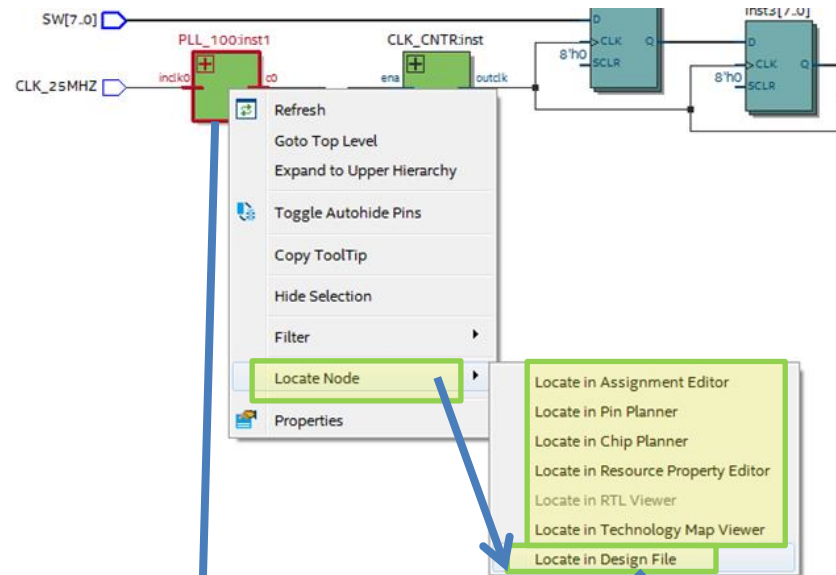


Рис. 47 Перекрестная ссылка из RTL Viewer в Schematic Editor

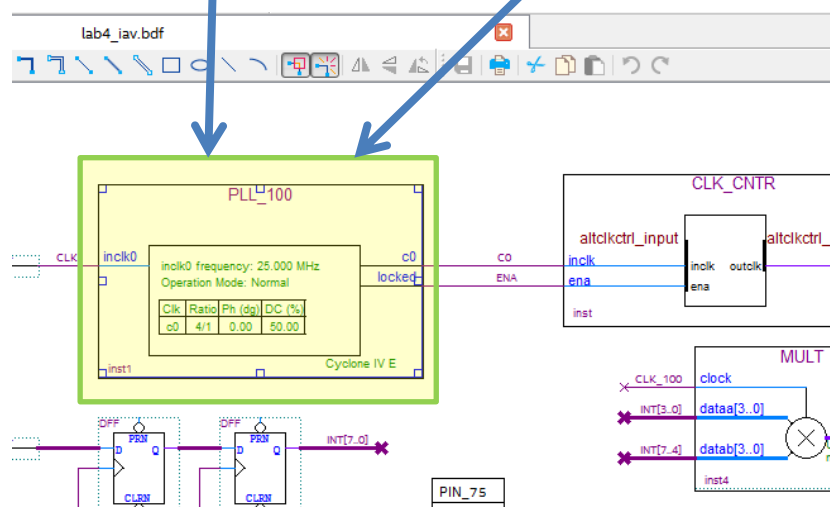


Рис. 48 Окно схематического редактора с выделенным в RTL Viewer компонентом

Таким образом, можно попасть из **RTL Viewer** в любой редактор (рис. 47).

9.7. Откройте нижний уровень детализации **bin\_7seg** в компоненте **ind\_drv** (рис. 43) и с помощью перекрестной ссылки перейдите в **Technology Map Viewer (Post-Mapping)**. Откроется окно **Technology Map Viewer** с реализацией компонента **bin\_7seg** в базе ПЛИС (на логических ячейках) (рис. 49). Сравните с представлением в **RTL Viewer**, сохраните оба скриншота для отчета.

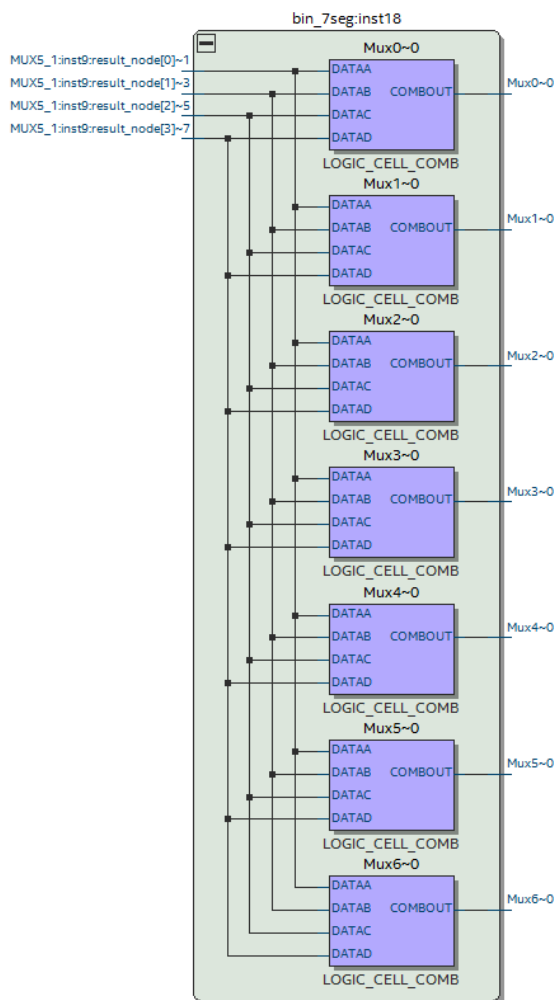


Рис. 49 Реализация компонента **bin\_7seg** на элементарном уровне в окне Technology Map Viewer

9.8. Выделите любой логический элемент **Logic\_Cell\_Comb (LE)** компонента **bin\_7seg** в **Technology Map Viewer** и с помощью перекрестной ссылки (рис. 50) откройте **Resource Property Editor** (рис. 51).

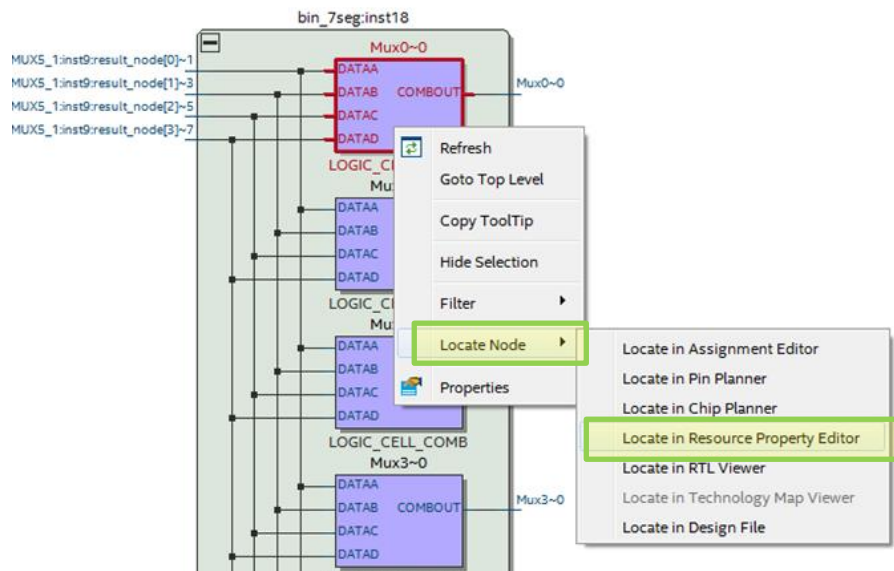


Рис. 50 Перекрестная ссылка из **Technology Map Viewer** в **Resource Property Editor**

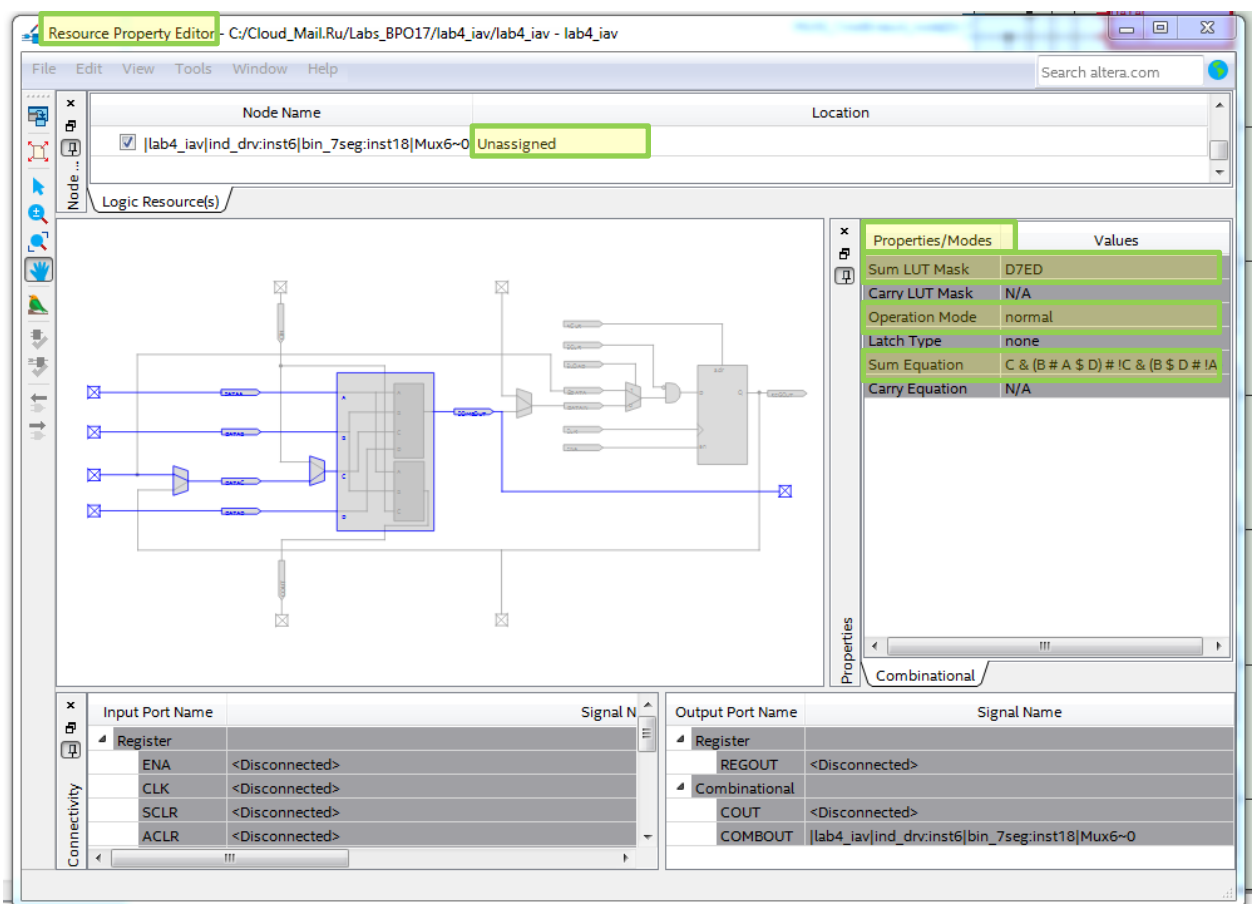


Рис. 51 Отображение части схемы проекта, реализуемой в логической ячейке, в **Resource Property Editor**

Будет отображаться структура выбранной логической ячейки на схемном уровне детализации. Синим цветом, выделены элементы, участвующие в реализации функции схемы. Остальные элементы не используются. Так как, полная компиляция еще не выполнялась, трассировки схемы на кристалле нет, и информация о конкретной



выбранной логической ячейке (в поле **Location** нет назначенных ресурсов: **Unassigned** ) на кристалле тоже нет. Но в окне **Properties/Modes** есть информация о реализованной в этой ячейке функции части схемы: в строке **Sum Equation** логическое выражение описывающее функцию, реализованной на этой логической ячейке **C&(B#A\$D) # !C&(B\$D#!A)** и в строке **Sum LUT Mask** таблица истинности ячейки памяти - **D7ED** (рис. 51). Также отображается режим работы логической ячейки – **normal** (обычный, в отличие от *арифметического* режима, используемого для реализации сумматоров).

9.9. Откройте компонент **MULT** в **RTL viewer** до самого нижнего уровня иерархии и, с помощью перекрестной ссылки, посмотрите реализацию элемента **mac\_mult1** (сам умножитель) в **Resource Property Editor**. Сделайте скриншот схемы элемента для отчета (рис. 52). Видно, что для реализации умножителя используется встроенный умножитель, а не логические ячейки.

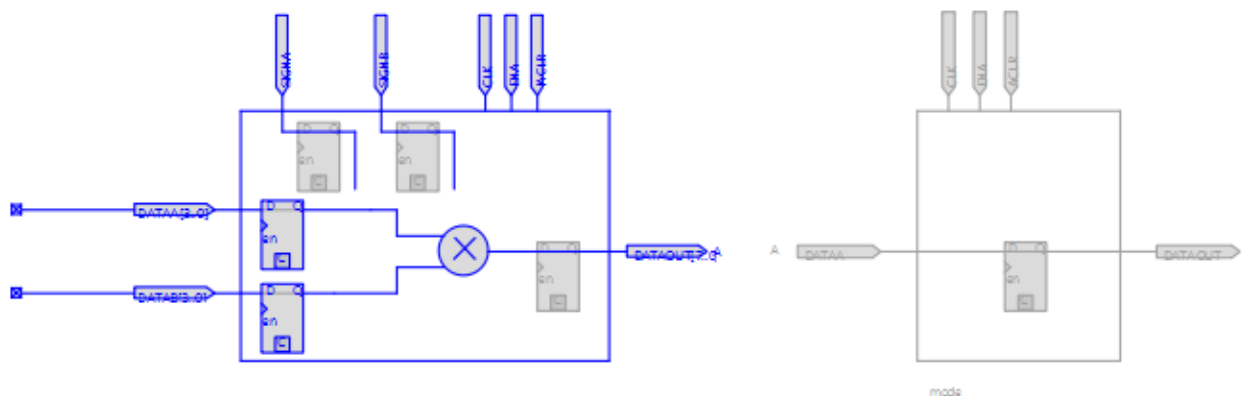


Рис. 52 Отображение схемы умножителя проекта, реализуемой с использованием встроенного умножителя, в **Resource Property Editor**

9.10. Любой из средств просмотра проекта можно открыть самостоятельно из вкладки **Tools**. Например, выполните **Tools=>Netlist Viewers=>Technology Map Viewer (Post-Mapping)**. Откроется окно **Technology Map Viewer**, где можно увидеть схематическое представление проекта в базе элементов ПЛИС (**Logic Cell**), но без привязки к конкретным ресурсам микросхемы (рис.53). Обратите внимание, что в окне редактора стали видны блоки ввода (IO\_IBUF) и вывода (IO\_OBUF).

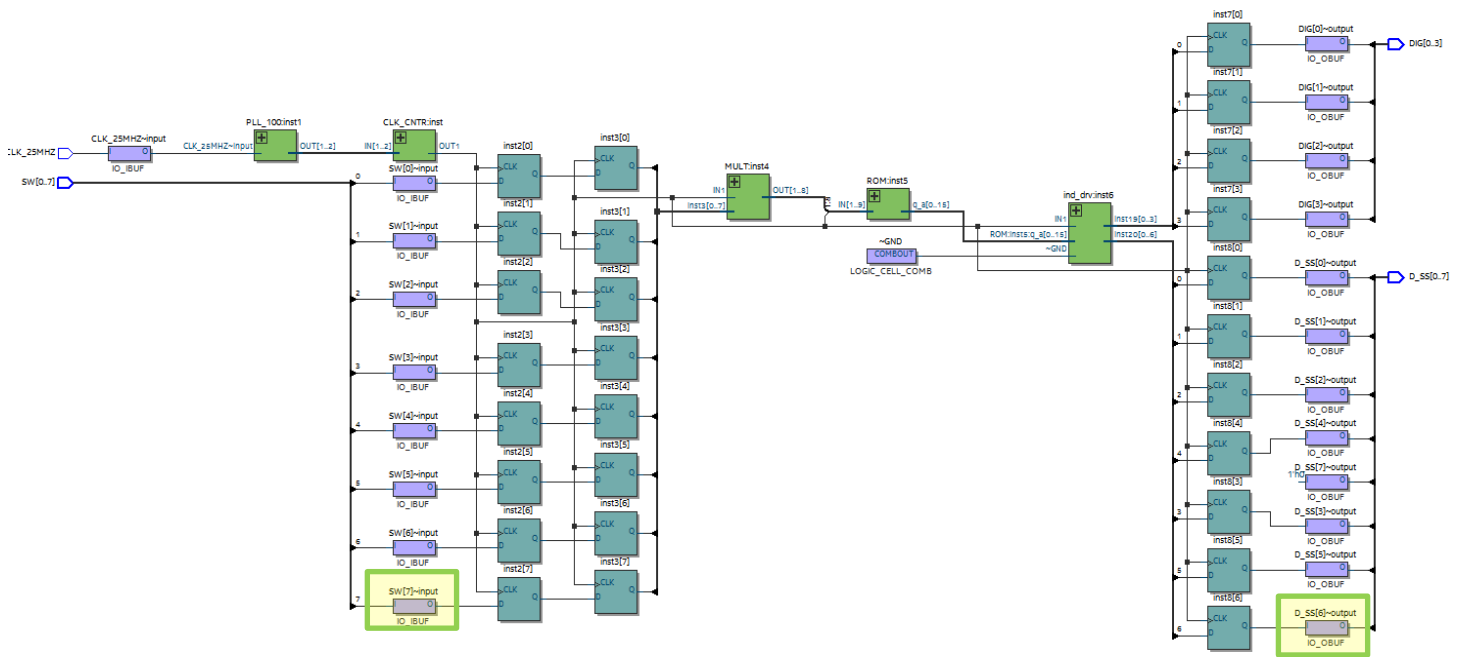


Рис. 53 Представление проектного файла в Technology Map Viewer (Post Mapping)

## 10. Назначение выводов ПЛИС

10.1. Откройте редактор **Pin Planner** одним из следующих способов: **Assignments -> Pin**

**Planner** / **Ctrl + Shift + N** / кнопка **Pin Planner**  на панели задач



10.2. В поле **Location** для каждого сигнала необходимо установить номер вывода микросхемы в соответствии со схемой отладочной платы. Для этого в строке выбранного сигнала в поле **Location** кликните ЛКМ и наберите номер вывода как на рис. 54 (набирайте только номер вывода!). Затем выполните **Enter** и выберите следующий сигнал. После того как все сигналы будут привязаны к выводам микросхемы ПЛИС, закройте окно редактора, все изменения автоматически сохранятся в файле проекта **lab4\_iav.qsf** (quartus setting file).

Named: * Edit: [X] [Y]			
Node Name	Direction	Location	I/O Bank
CLK_25MHZ	Input	PIN_23	1
D_SS[7]	Output	PIN_75	5
D_SS[6]	Output	PIN_84	5
D_SS[5]	Output	PIN_76	5
D_SS[4]	Output	PIN_85	5
D_SS[3]	Output	PIN_77	5
D_SS[2]	Output	PIN_86	5
D_SS[1]	Output	PIN_133	8
D_SS[0]	Output	PIN_87	5
DIG[3]	Output	PIN_73	5
DIG[2]	Output	PIN_80	5
DIG[1]	Output	PIN_74	5
DIG[0]	Output	PIN_83	5
SW[7]	Input	PIN_88	5
SW[6]	Input	PIN_89	5
SW[5]	Input	PIN_90	6
SW[4]	Input	PIN_91	6
SW[3]	Input	PIN_49	3
SW[2]	Input	PIN_46	3
SW[1]	Input	PIN_25	2
SW[0]	Input	PIN_24	2

Рис.54 Окно редактора **Pin Planner** с привязкой сигналов проекта к выводам ПЛИС

10.3. Все неиспользованные в проекте выводы микросхемы ПЛИС необходимо установить в режим «**AS INPUT TRI-STATED**». Для этого выполните **Assignments->Device->Device and Pin Options**. В появившемся окне **Device and Pin Options** выберите в поле **Category** строку **Unused Pins** и в окошке **Reserve all unused pins** выберите **As input tri-stated with weak pull-up**. После назначения выводов у символов ввода-вывода на схеме появятся номера выводов (**PIN\_nn**) микросхемы, к которым они привязаны (рис. 33).

## 11. Создание sdc – файла и выполнение полной компиляции проекта

11.1. Создайте файл с информацией о временных требованиях (т.е. на какой частоте проект должен работать гарантированно) к проекту:

**File =>New => Other Files => Synopsys Design Constraints File**

11.2. Сохраните созданный текстовый проектный файл **File =>Save As** с именем **lab4\_iav.sdc**. Минимальный набор команд, который должен быть в нем определен, показан на рисунке 55.

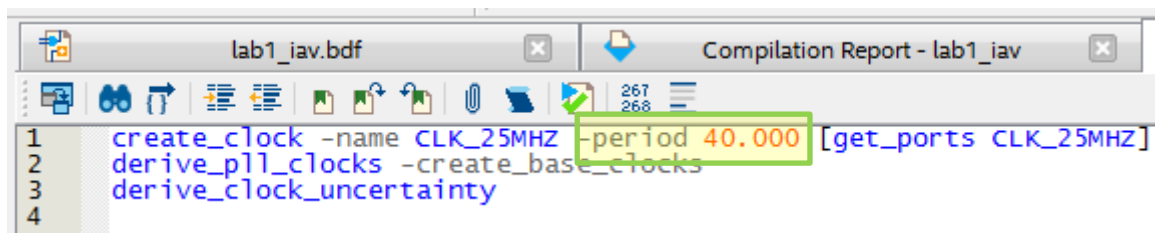


Рис. 55 Задание временных требований к проекту

Скопируйте и внесите в файл следующие строки (чтобы не набирать вручную):

```
create_clock -name CLK_25MHZ -period 40.000 [get_ports CLK_25MHZ]
derive_pll_clocks -create_base_clocks
derive_clock_uncertainty
```

В первой строке задана частота, которой должен удовлетворять проект. Данная частота (25MHz) поступает на вход ПЛИС с выхода генератора прямоугольных сигналов отладочной платы.

**Подсказка.**

Откройте *lab1\_iav.sdc* из проекта *lab1\_iav* (File-> Open-> <path project/ lab1\_iav/...>). Сохраните под именем *lab4\_iav.sdc* в папку проекта *lab4\_iav* и добавьте в текущий проект *lab4\_iav*. Важно, чтобы имена сигналов, для которых определены временные требования, в обоих проектах совпадали. При необходимости, отредактируйте значения временных параметров и имена сигналов.

11.3. Сохраните файл и проследите, чтобы файл появился в окне **Project Navigator** (Выберите вкладку **Files**). В противном случае, при открытом окне с файлом *lab4\_iav.sdc* выполните **Project=>Add Current File to Project**. Файл *lab4\_iav.sdc* добавится в проект.

11.4. После назначения сигналов на выводы микросхемы необходимо выполнить разводку схемы на кристалле ПЛИС (**Fitter**). Для этого выберите задачу **Compile Design**

из окна задач **Tasks** или нажмите кнопку  на панели задач



или выполните **Ctrl+L** или

**Processing->Start Compilation**. Запустится компиляция, результаты которой можно наблюдать в окне **Compilation Report** (рис. 56). В окне отображаются основные параметры проекта и затраченные на его реализацию ресурсы микросхемы (количество логических элементов и использованных в них триггеров, количество выводов микросхемы, количество встроенных блоков памяти, умножителей и PLL). Обратите внимание, что в данном проекте используются, кроме логических ячеек, один встроенный блок памяти, один умножитель частоты (PLL) и один умножитель

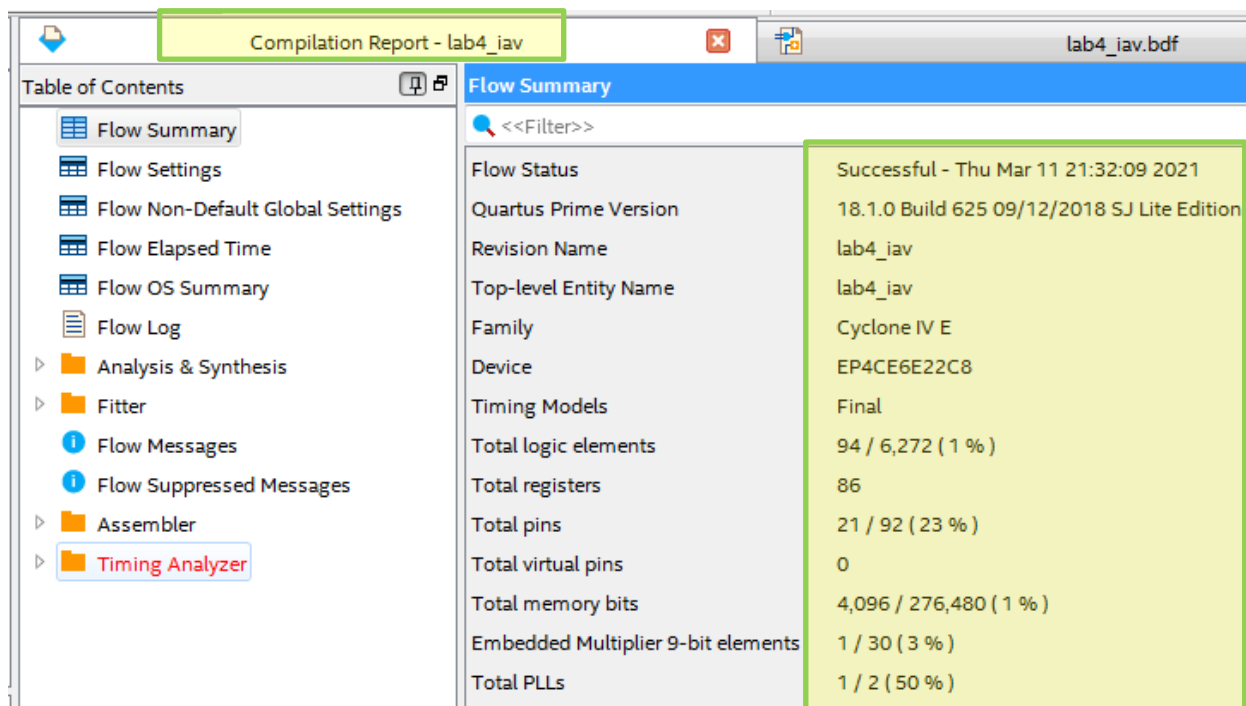


Рис. 56 Окно **Compilation Report** с результатами полной компиляции

11.5. В случае успешной компиляции в окне **Message** появится информация об ее успешном окончании. В противном случае, устраните ошибки.

11.6. В окне **Compilation Report** откройте файл **Fmax Summary** с результатами временного анализа (рис. 57).

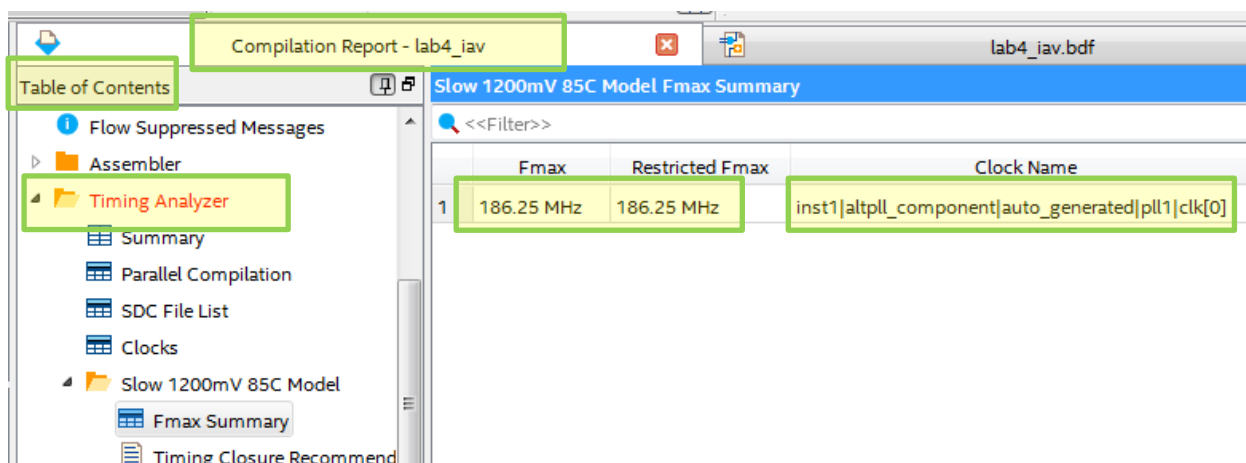


Рис. 57 Окно **Compilation Report** с результатами временного анализа проекта

Убедитесь, что максимально возможная частота работы проекта (Fmax), превышает заданные в файле **lab4\_iav.sdc** временные требования. В данном проекте временной анализ работы схемы производится относительно частоты с выхода PLL равной 100MHZ.

## 12. Анализ проекта с помощью Technology Map Viewer (Post-Fitting) и State Machine Viewer

12.1. После выполнения полной компиляции проекта можно посмотреть реализацию схемы с учетом привязки к конкретным ресурсам ПЛИС.

Выполните **Tools=>Netlist Viewers=>Technology Map Viewer (Post-Mapping)**. Откроется окно **Technology Map Viewer**, где можно увидеть схематическое представление проекта в базе элементов ПЛИС (**Logic Cell**), с привязкой к конкретным ресурсам микросхемы (рис.58). Показаны все задействованные логические ячейки и элементы ввода и вывода, а также все служебные сигналы, участвующие в реализации схемы. В частности показаны выходы ПЛИС двойного назначения, которые используются для конфигурирования микросхемы на отладочной плате.

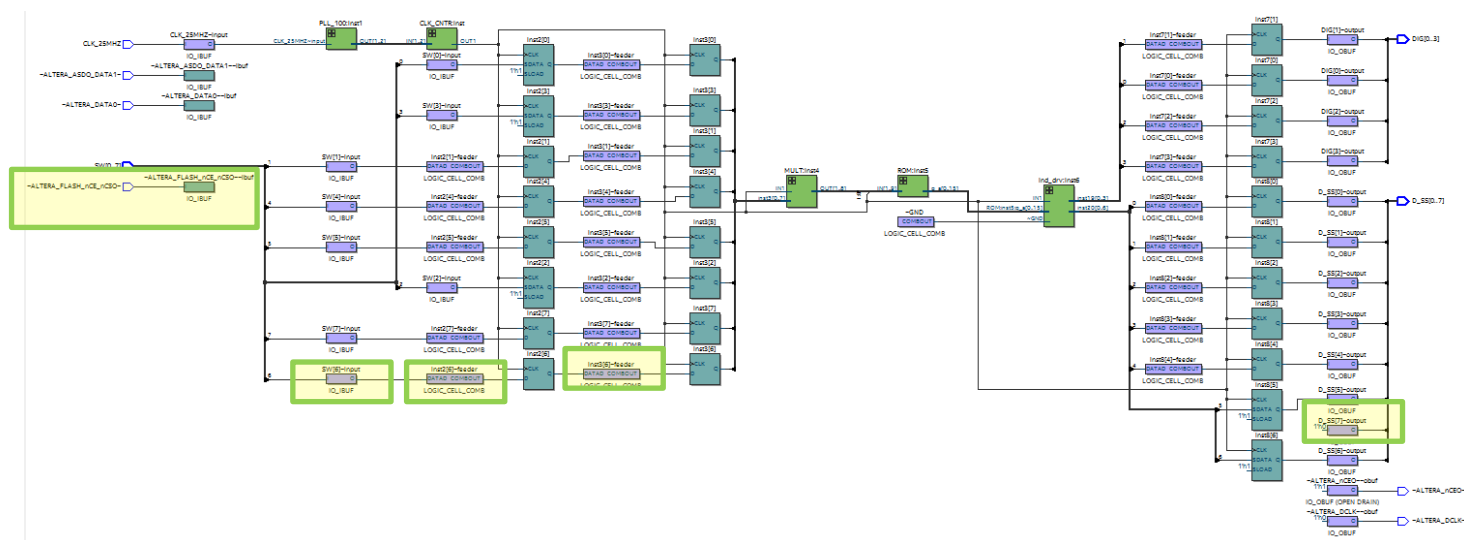


Рис. 58 Представление проектного файла в Technology Map Viewer (Post Fitting)

12.2. Откройте нижний уровень детализации **bin\_7seg** в компоненте **ind\_drv**. Выделите любой логический элемент **Logic\_Cell\_Comb (LE)** компонента **bin\_7seg** и с помощью перекрестной ссылки откройте **Resource Property Editor** (рис. 59). Дополнительно к информации, которую можно было посмотреть в **Technology Map Viewer (Post-Mapping)** после анализа и синтеза, в поле **Location** добавилась информация о номере ячейки на которой реализуется эта часть схемы: **LCCOMB\_X31\_Y9\_N22**.

12.3. Откройте компонент **MULT** до самого нижнего уровня иерархии и, с помощью перекрестной ссылки, посмотрите реализацию элемента **mac\_mult1** (сам умножитель) в **Resource Property Editor**. Сделайте скриншот схемы элемента для отчета (рис. 60). Видно, что для реализации умножителя появилась информация о координатах умножителя на кристалле и его номер **DSPMULT\_X20\_Y7\_N0**.

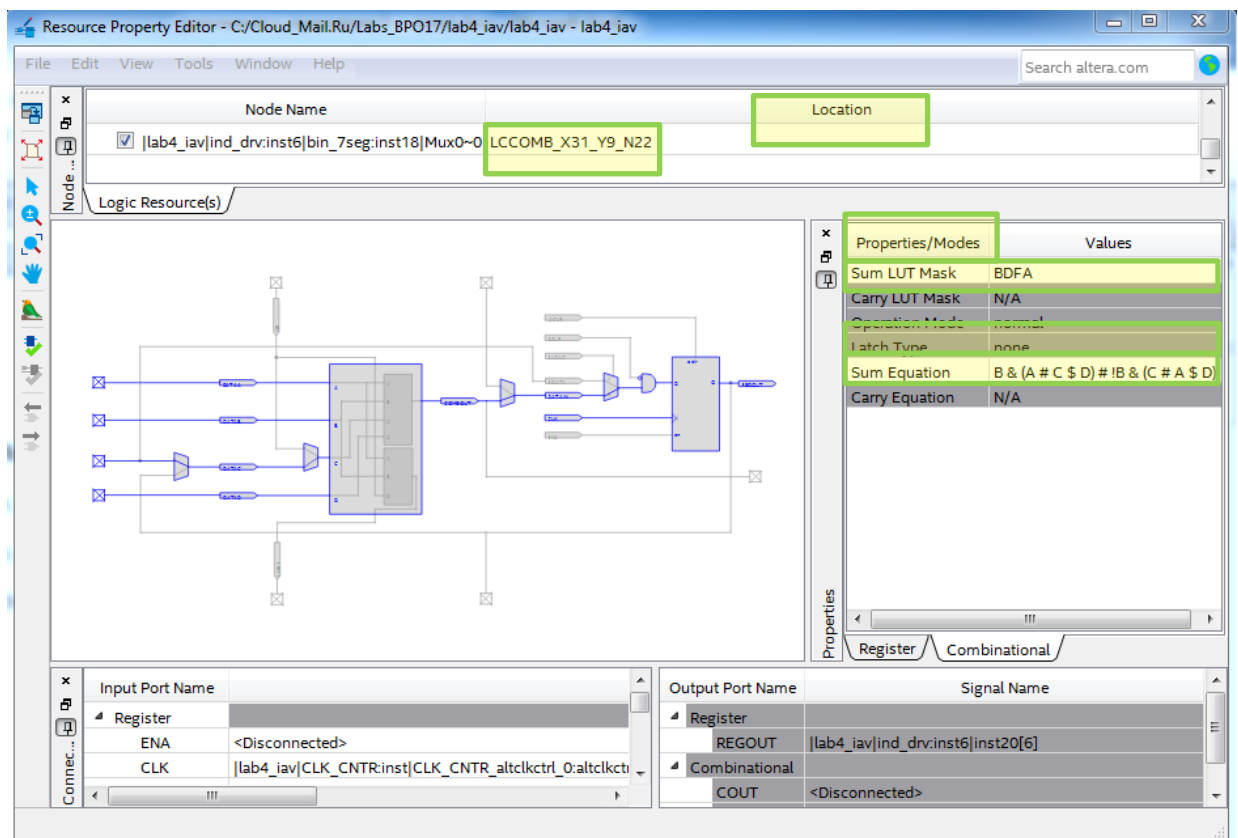


Рис. 59 Отображение части схемы проекта, реализуемой в логической ячейке, в **Resource Property Editor**

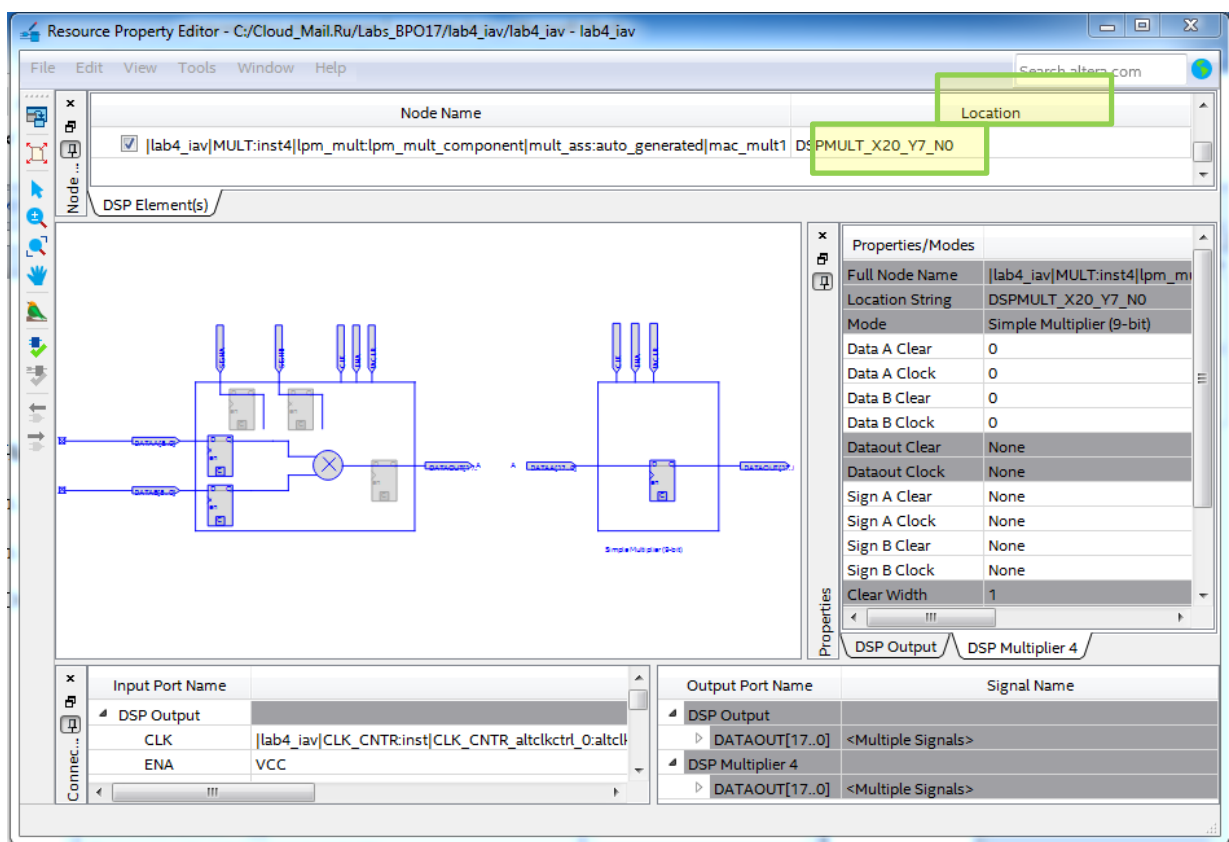


Рис. 60 Отображение схемы умножителя проекта, реализуемой с использованием встроенного умножителя, в **Resource Property Editor**

12.4. Выполните **Tools=>Netlist Viewers=>State Machine Viewer**. Откроется окно **State Machine Viewer**, где можно увидеть схематическое представление конечных автоматов (КА), если они присутствуют в проекте. В данном проекте есть один КА в компоненте **ind\_drv** (рис. 61).

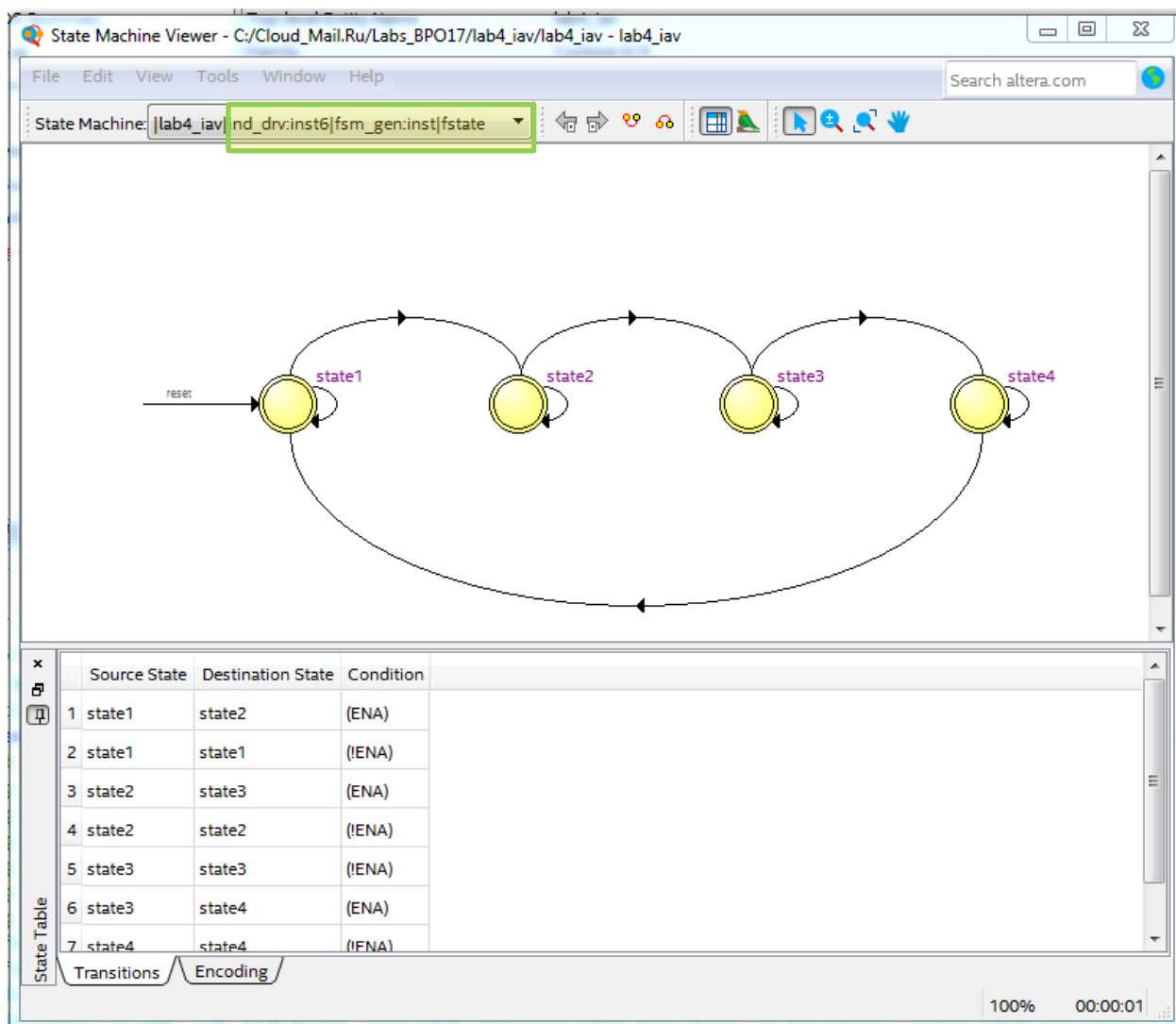


Рис. 61 Отображение конечного автомата с помощью **State Machine Viewer**

На вкладках **Transitions** и **Encoding** можно посмотреть таблицу переходов между состояниями конечного автомата и выбранные компилятором коды для состояний автомата. С помощью перекрестных ссылок можно посмотреть реализацию КА в базисе ПЛИС, например, в **Technology Map Viewer (Post-Fitting)** (рис. 62 )



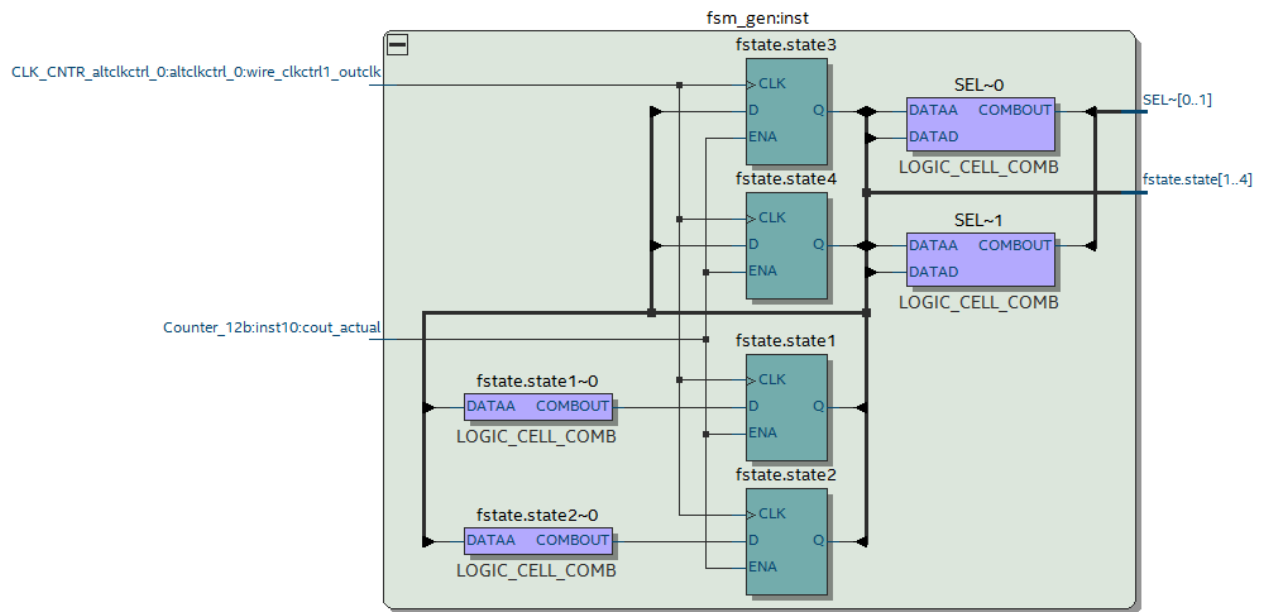


Рис. 62 Отображение реализации конечно автомата в схеме компонента `ind_drv` с помощью **Technology Map Viewer**

### 13. Анализ проекта с помощью Chip Planner

13.1. Выполните **Tools=>Chip Planner**. Откроется окно редактора с представлением проекта на кристалле (рис. 63).

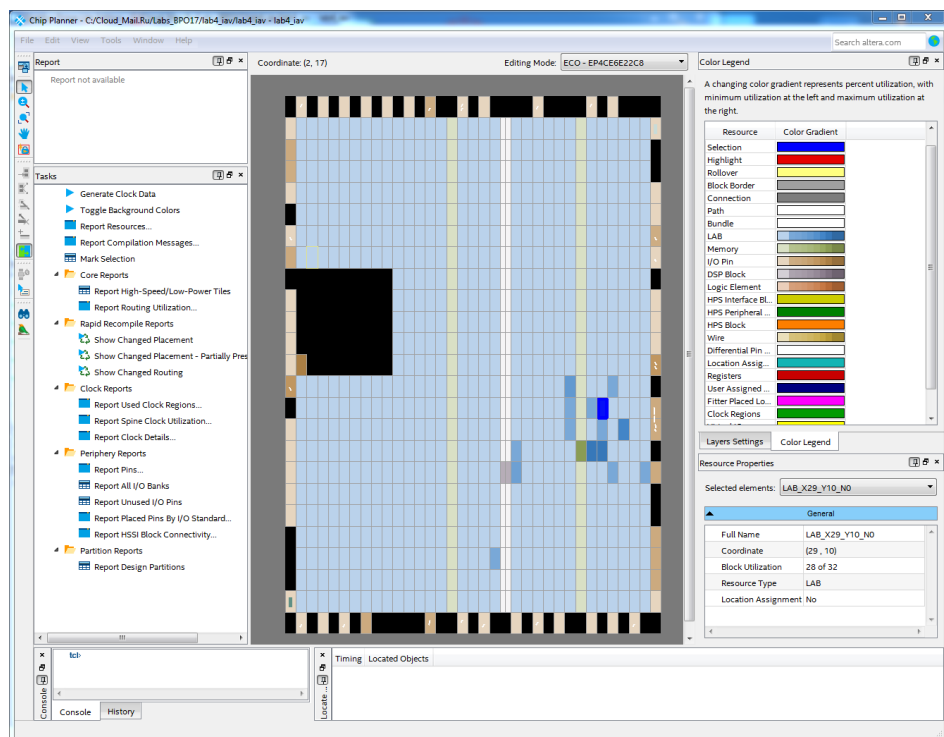


Рис. 63 Окно редактора Chip planner

13.2. Выполните задачу **Report Resource...** в разделе **Tasks**. Для этого дважды кликните левой кнопкой мыши на строку **Report Resource** или кликните правой кнопкой мыши на строку **Report Resource** и в выпавшем меню выберите **Start**. Затем в выпавшем окне **Report Resource** выберите в меню **Resource Type** строку с элементом **PLL** и выполните «Ok» (рис. 64). Выбранный элемент отобразится в редакторе в центральном окне на символе кристалла ПЛИС. В данной микросхеме два элемента **PLL**. Они отобразятся в правом верхнем углу и нижнем левом углах (рис. 65).

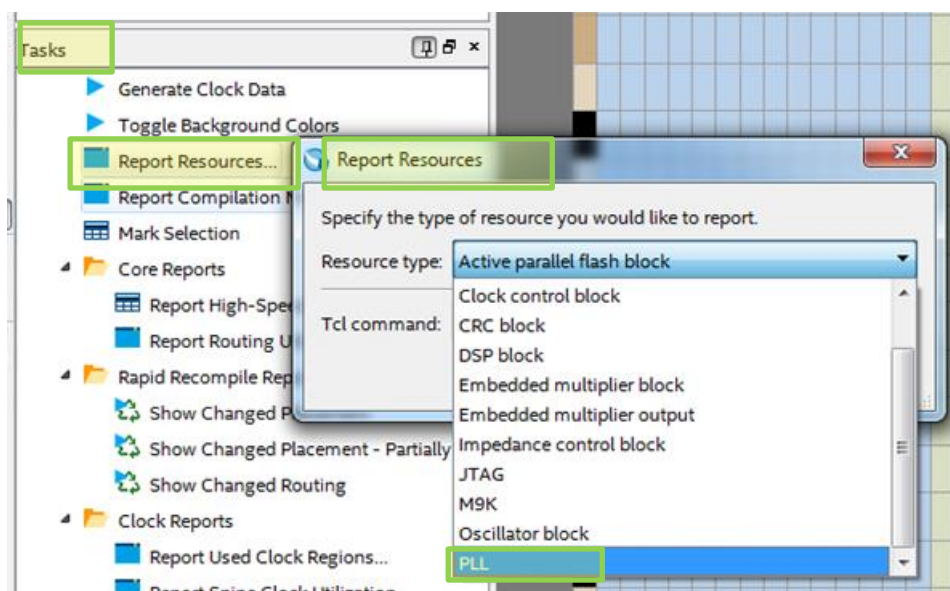



Рис. 64 Выбор элементов схемы проекта для отображения на кристалле



Рис. 65 Отображение элементов PLL на символе кристалла ПЛИС

13.3. Аналогично выберите для отображения элементы **DSP Block**, **M9K** модули памяти, **Clock Control Block** (рис. 66).

13.4. Кликните ЛКМ на символ **DSP Block**, затем увеличьте его с помощью инструмента  на левой боковой панели инструментов или **Ctrl**+колесо мыши. Кликните ЛКМ на

одну из крайних блоков символа. В окошке **Selected Elements** появится реализация умножителя как в редакторе **Resource Editor** и в окне **Properties/Modes** подробная информация о выбранном элементе (рис. 67) Количество регистров в умножителе равно 2, что соответствует настройке умножителя с 2-мя тактами конвейеризации при его создании.

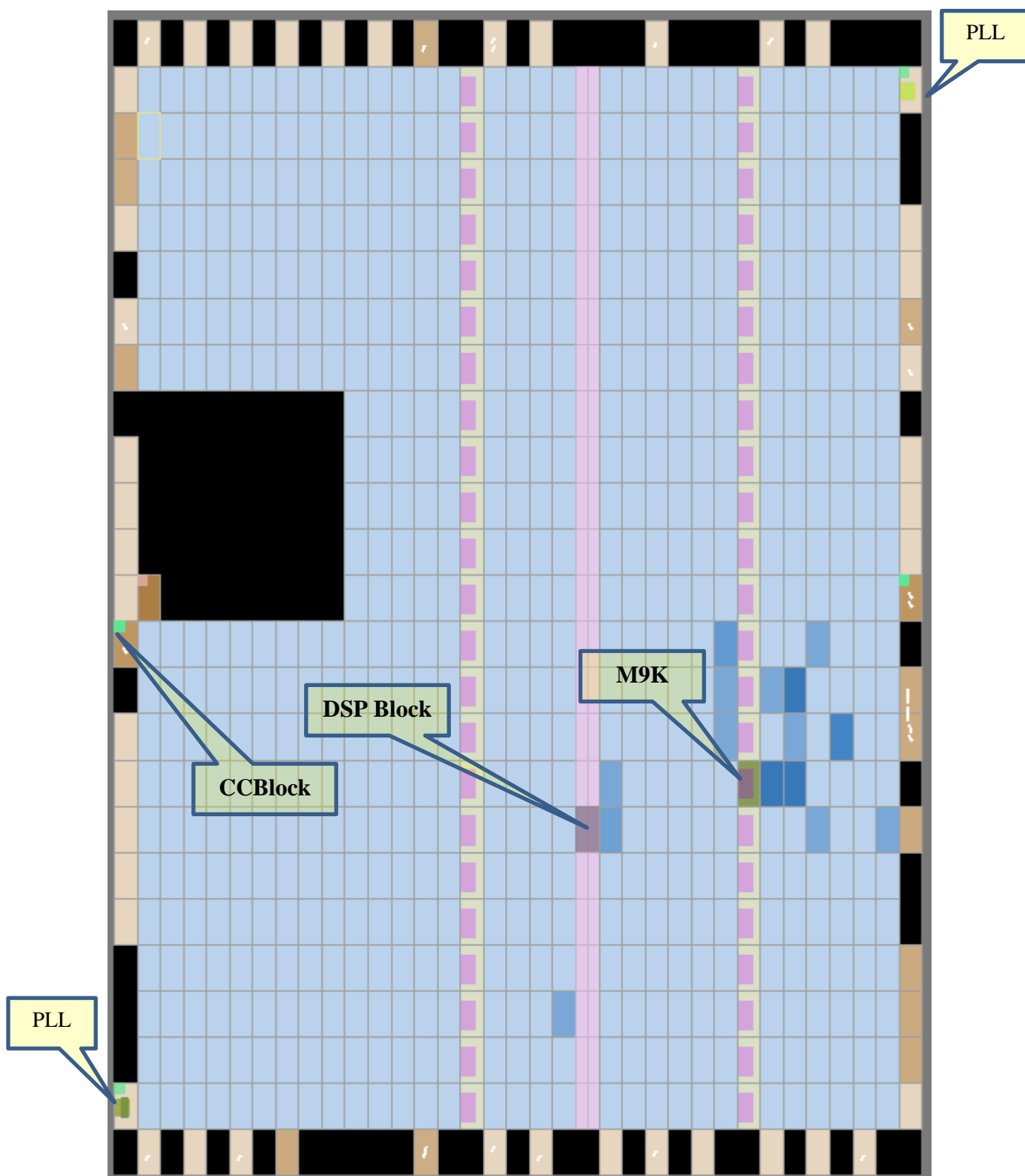


Рис. 66 Отображение выбранных элементов схемы на символе кристалла ПЛИС

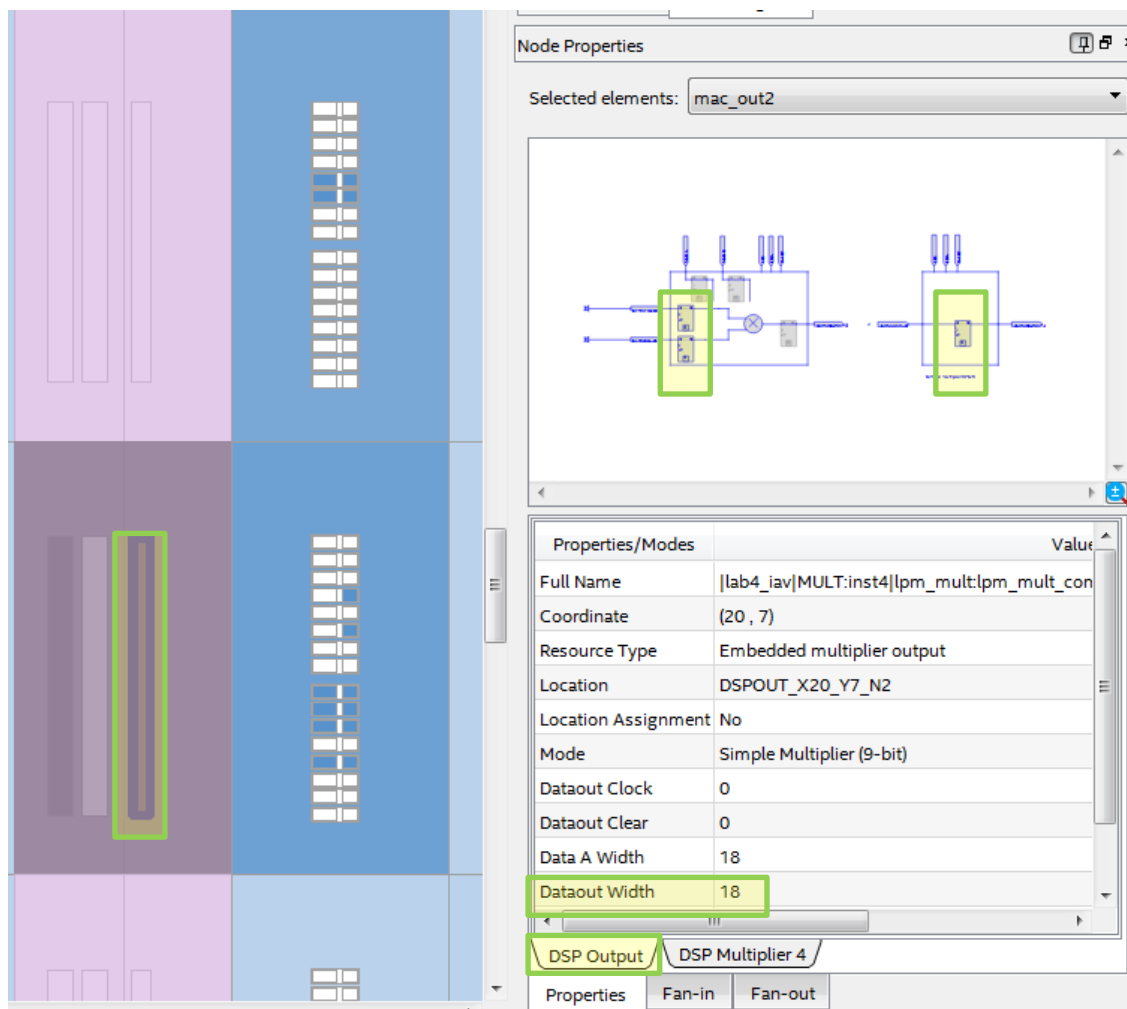


Рис. 67 Детальное отображение элемента DSP Block на символе кристалла ПЛИС

Дважды кликните на поле со схемой модуля, откроется окно редактора **Resource Properties Editor**, аналогичное тому, что открывали в разделе 12. Закройте этот редактор.

13.5. Выполните **View=>Fit in Window**. Выберите и просмотрите аналогично модуль памяти **M9K**. Сохраните результат просмотра для отчета (аналогично рис. 68).

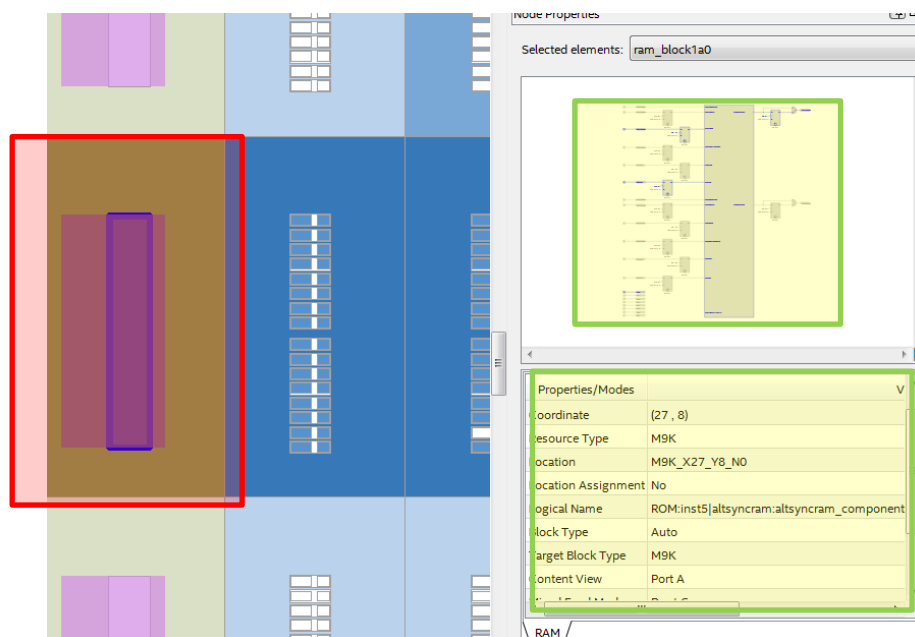



Рис. 68 Детальное отображение элемента M9K на символе кристалла ПЛИС

13.6. Выполните **View=>Fit in Window**. Выберите **PLL** в нижнем левом углу (он используется в проекте). Затем последовательно нажмите инструмент  на левой панели инструментов 3 раза:

после первого нажатия появится связь от **PLL** к **Control Clock Block** (рис. 69a);

после второго нажатия будет показан переход внутри модуля **Control Clock Block** (рис. 69b);

после третьего нажатия будут показаны тактовые сигналы, поступающие на все триггеры схемы проекта. переход внутри модуля **Control Clock Block** (рис.69с).

Видно, что тактовые сигналы не поступают в блоки ввода и вывода микросхемы. Следовательно, в них не используются имеющиеся там триггеры. Чтобы убедиться в этом откройте Pin Planner, выберите сигнал SW[2] и с помощью перекрестной ссылки откройте окно редактора **Resource Properties Editor** (рис. 70). Видно, что сигнал с вывода микросхемы проходит внутрь кристалла, минуя входной триггер (рис. 71). Сохраните скриншот для отчета.

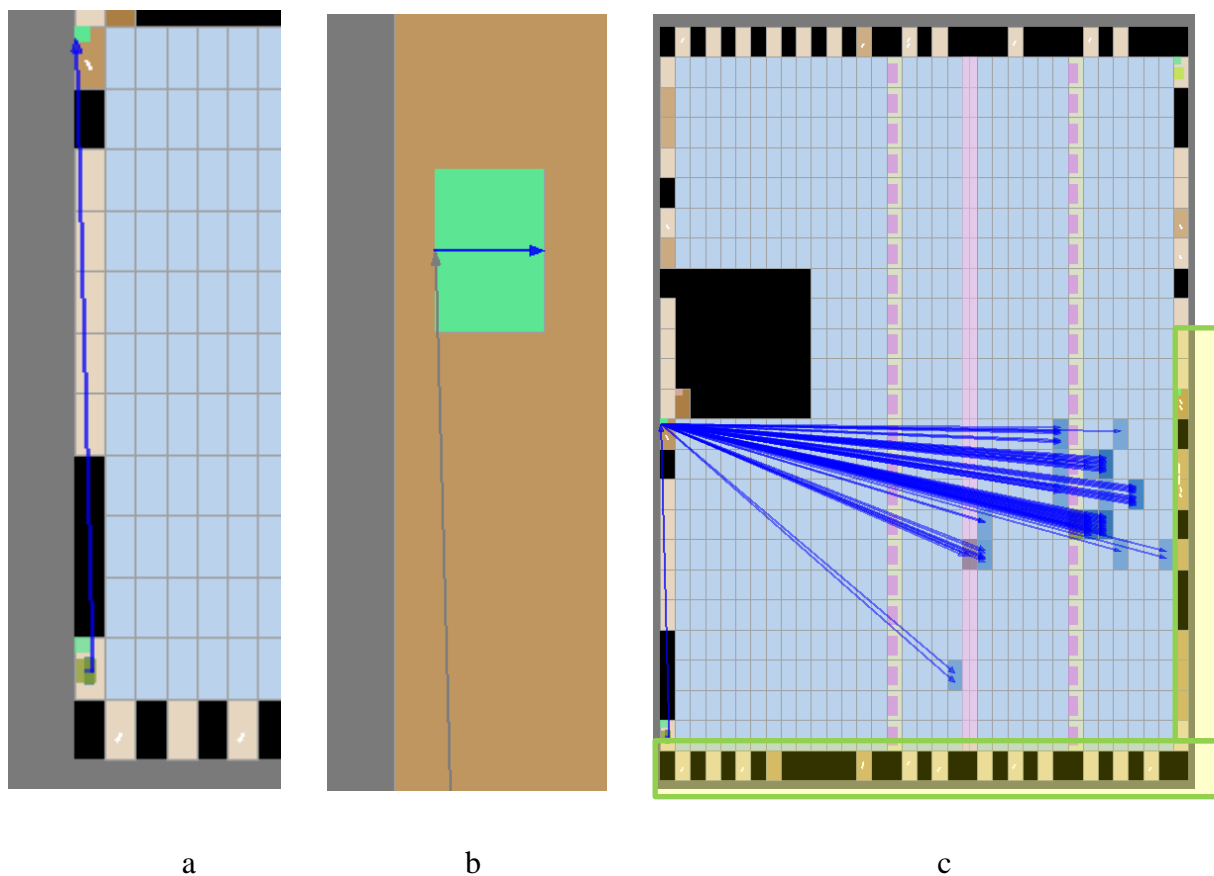


Рис. 69 Просмотр прохождения сигналов тактирования внутри кристалла

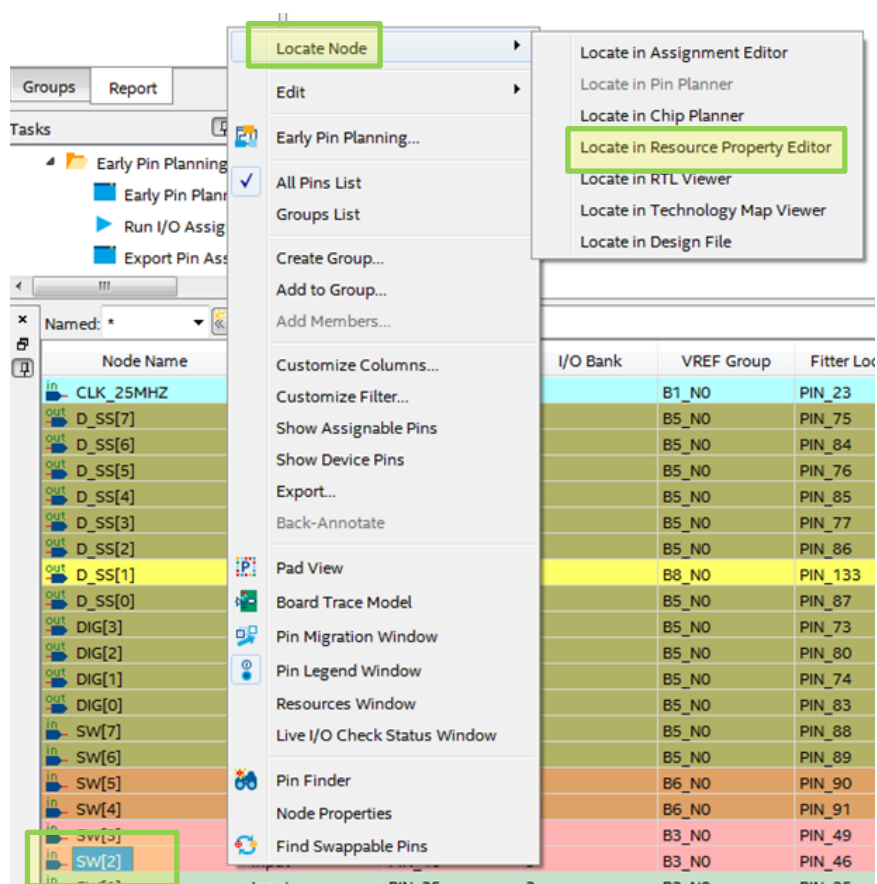


Рис. 70 Просмотр прохождения сигнала через вывод поступления сигнала SW2

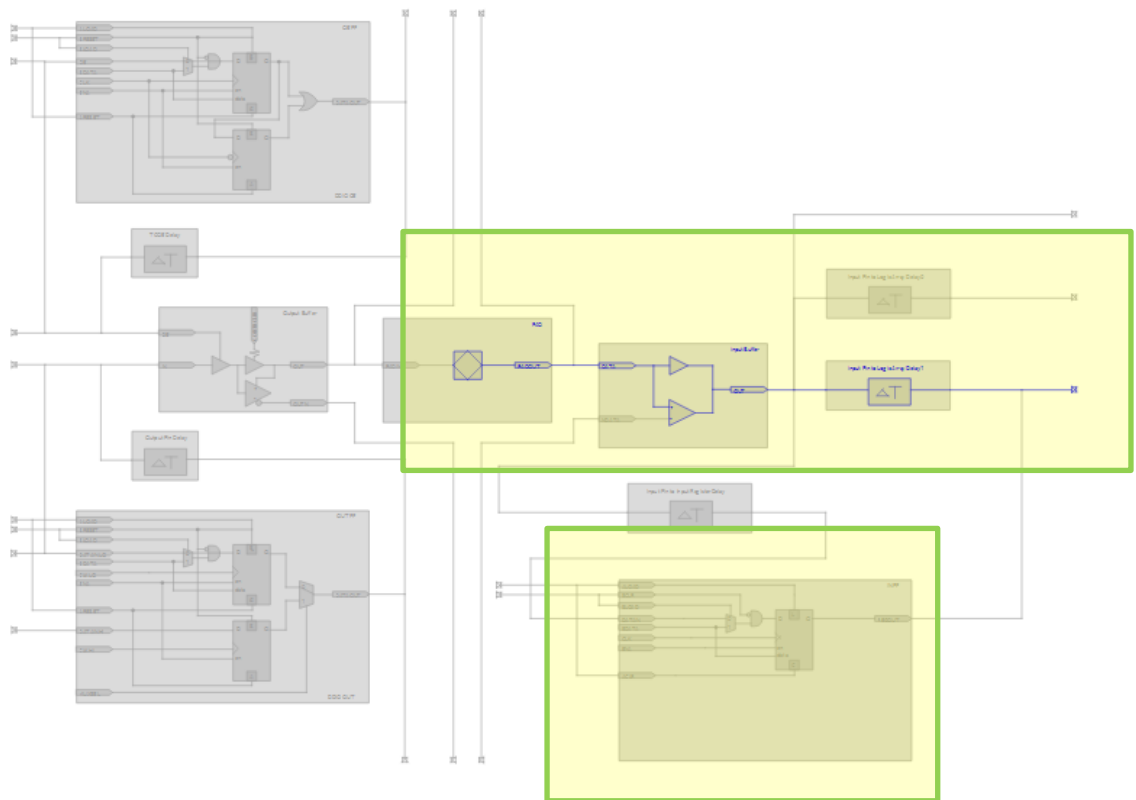


Рис. 71 Просмотр прохождения сигнала SW2 через цепи блока ввода-вывода микросхемы

## 14. Управление компиляцией проекта с помощью Assignment Editor

14.1. Выделите группу сигналов **SW[7..0]** в схеме проекта **lab4\_<fio>.bdf** и с помощью перекрестных ссылок (рис.72) откройте редактор назначений **Assignment Editor** (рис.73).

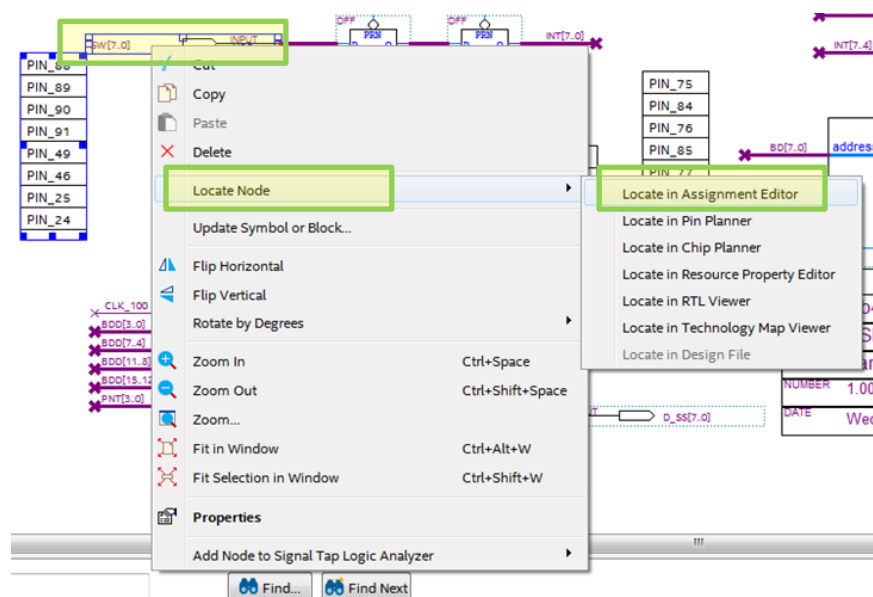


Рис. 72 Выбор сигналов SW[7..0] и открытие редактора Assignment Editor

tatu	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✓	in SW[7]	Location	PIN_88	Yes			
2	✓	in SW[6]	Location	PIN_89	Yes			
3	✓	in SW[5]	Location	PIN_90	Yes			
4	✓	in SW[4]	Location	PIN_91	Yes			
5	✓	in SW[3]	Location	PIN_49	Yes			
6	✓	in SW[2]	Location	PIN_46	Yes			
7	✓	in SW[1]	Location	PIN_25	Yes			
8	✓	in SW[0]	Location	PIN_24	Yes			
9	<<new>>	<<new>>	<<new>>					

Рис. 73 Окно редактора Assignment Editor

14.2. В столбце «**To**» дважды кликните по полю «**new**» и из выпавшего списка выберите группу **SW** (рис.74).

tatu	From	To	Assignment Name
1	✓	in SW[0]	Location
2	✓	in SW[1]	Location
3	✓	in SW[2]	Location
4	✓	in SW[3]	Location
5	✓	in SW[4]	Location
5	✓	in SW[5]	Location
7	✓	in SW[6]	Location
8	✓	in SW[7]	Location
9	<<new>>	SW	<<new>>

Рис. 74 Выбор сигнала SW для назначения необходимых опций в редакторе Assignment Editor

14.3. В столбце «**Assignment Name**» дважды кликните по полю «**Location**» в строке с сигналом **SW** и из выпавшего списка выберите опцию **Fast Input Register** (рис.75).

14.4. В столбце «**Value**» дважды кликните по полю «**пустое**» в строке с сигналом **SW** и из выпавшего списка выберите опцию «**On**». Окончательный вид настроек для сигнала **SW** представлен на рис. 76. В колонке «**Enabled**» должно быть значение «**Yes**».



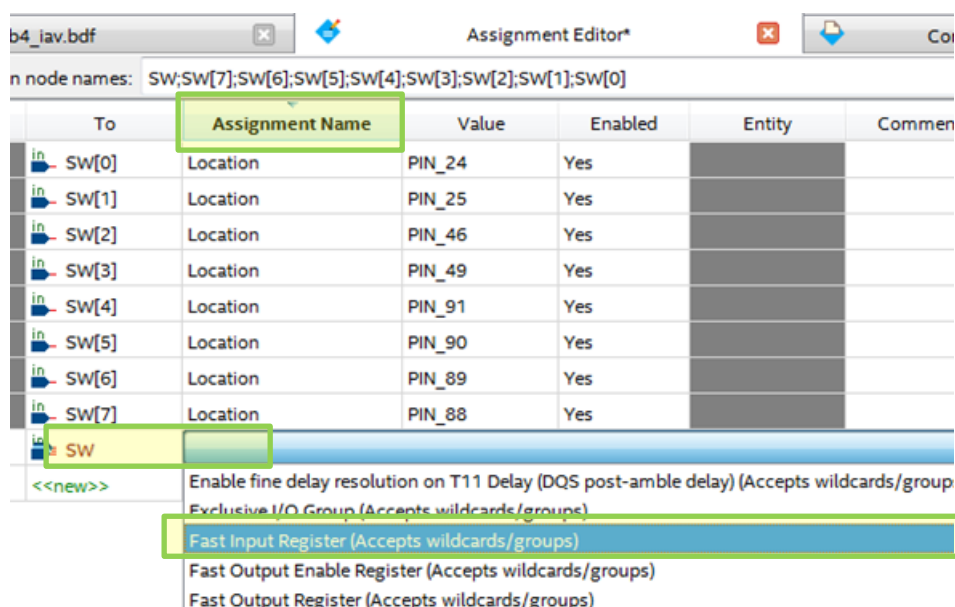




Рис. 75 Выбор опции Fast Input Register для сигналов SW

	tatu	From	To	Assignment Name	Value	Enabled
1	✓		in SW[0]	Location	PIN_24	Yes
2	✓		in SW[1]	Location	PIN_25	Yes
3	✓		in SW[2]	Location	PIN_46	Yes
4	✓		in SW[3]	Location	PIN_49	Yes
5	✓		in SW[4]	Location	PIN_91	Yes
6	✓		in SW[5]	Location	PIN_90	Yes
7	✓		in SW[6]	Location	PIN_89	Yes
8	✓		in SW[7]	Location	PIN_88	Yes
9	✓		in SW	Fast Input Register	On	Yes
10		<<new>>	<<new>>	<<new>>		

Рис. 76 Окончательный вид настроек для сигнала SW

14.5. В столбце «**From**» дважды кликните по полю «**new**» и запустите **Node Finder**

, далее нажмите кнопку  чтобы раскрыть все опции окна (если они закрыты). В строке «**Filter**» выберите «**Pins: output**». Затем выполните **List** и из окна **Matching Nodes** перенесите групповой сигнал **D\_SS** в окно **Node Found** (рис. 77). Выполните «**Ok**».

В столбце «**Assignment Name**» дважды кликните по полю «**Location**» в строке с сигналом **D\_SS** и из выпавшего списка выберите опцию **Fast Output Register** (рис.75).

В столбце «**Value**» для сигнала **D\_SS** выберите опцию «**On**». В колонке «**Enabled**» должно быть значение «**Yes**».

14.6. Настройте сигнал **DIG** аналогично **D\_SS**. Окончательный вид окна редактора назначений **Assignment Editor** показан на рис. 77. Сохраните назначения и выполните полную компиляцию проекта.

<<new>>		Filter on node names: SW;SW[7];SW[6];SW[5];SW[4];SW[3];SW[2];SW[1];SW[0]					
	tatu	From	To	Assignment Name	Value	Enabled	Entity
1	✓		in SW[0]	Location	PIN_24	Yes	
2	✓		in SW[1]	Location	PIN_25	Yes	
3	✓		in SW[2]	Location	PIN_46	Yes	
4	✓		in SW[3]	Location	PIN_49	Yes	
5	✓		in SW[4]	Location	PIN_91	Yes	
6	✓		in SW[5]	Location	PIN_90	Yes	
7	✓		in SW[6]	Location	PIN_89	Yes	
8	✓		in SW[7]	Location	PIN_88	Yes	
9	✓		in SW	Fast Input Register	On	Yes	lab4_iav
10	✓		out D_SS	Fast Output Register	On	Yes	lab4_iav
11	✓		out DIG	Fast Output Register	On	Yes	lab4_iav
12		<<new>>	<<new>>	<<new>>			

Рис. 77 Окончательный вид настроек для всех сигналов проекта

14.7. В окне **Message** для сигналов **SW[0]**, **SW[1]**, **SW[7..4]** будут сгенерированы сообщения (рис. 78). Далее мы увидим, как они связаны с результатами компиляции.

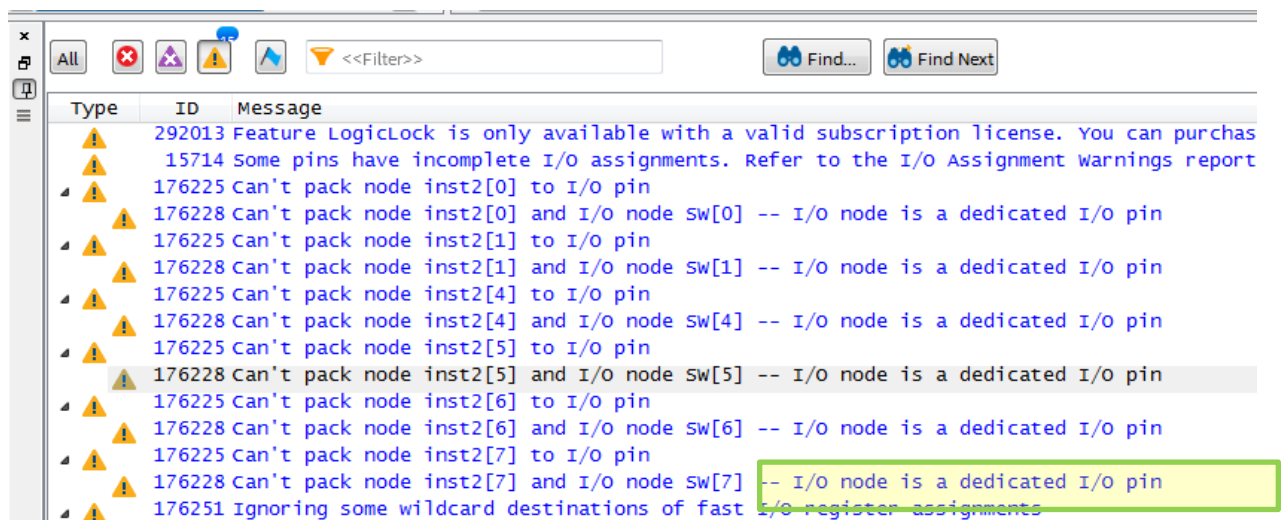


Рис. 78 Сообщения о невозможности выполнения назначения пользователя для сигналов SW[0], SW[1], SW[7..4]

14.8. Выберите сигнал **SW[7..0]**, нажмите правую кнопку мыши и выполните **Locate=> Locate in Pin Planner**.

Группа сигналов **SW** будет выделена в колонке **Node Name**.

14.9. Выберите сигнал **SW[2]** (pin 46), нажмите правую кнопку мыши и выполните **Locate=> Locate in Resource Properties Editor**. Видно, что сигнал с вывода **SW[2]** проходит к внутренним цепям микросхемы через триггер блока ввода-вывода (рис. 79)

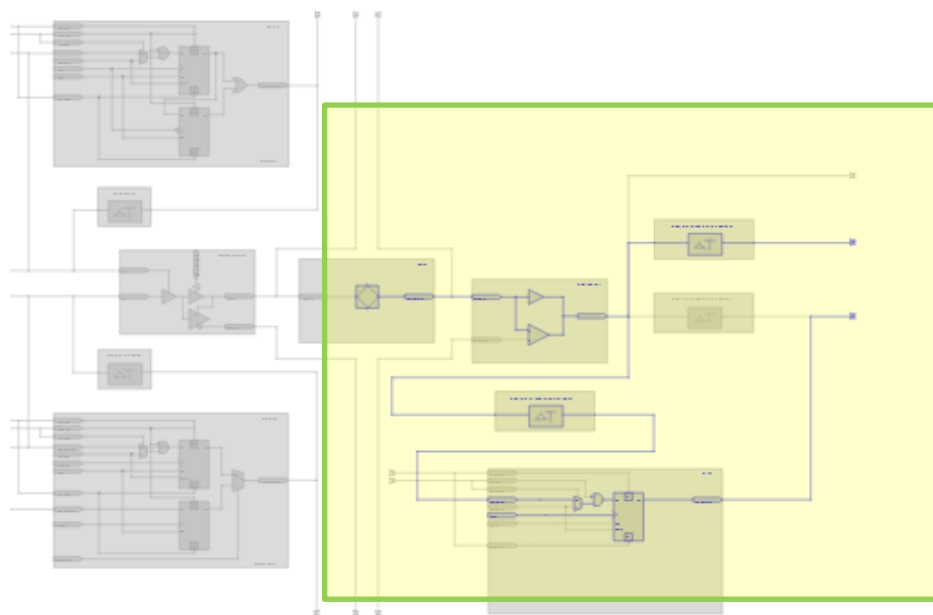


Рис. 79 Просмотр прохождения сигнала SW2 через цепи блока ввода-вывода микросхемы

14.10. Выберите сигнал **SW[0]** (pin 24) (он относится к группе сигналов, для которых есть предупреждения в окне **Message** после компиляции) в **Pin Planner**, нажмите правую кнопку мыши и выполните **Locate=> Locate in Resource Properties Editor**. Видно, что сигнал с вывода **SW[0]** проходит в внутренние цепи микросхемы напрямую, минуя все дополнительные цепи (рис. 80). Это связано с тем, что этот сигнал относится к группе **Dedicated** (выделенных сигналов специального назначения, в данном случае предназначенных для ввода тактовых сигналов). Это можно увидеть, если подвести курсор к выводу 24 на символе микросхемы в окне **Pin Planner**, при этом появится окошко с информацией о выводе (Рис. 81).

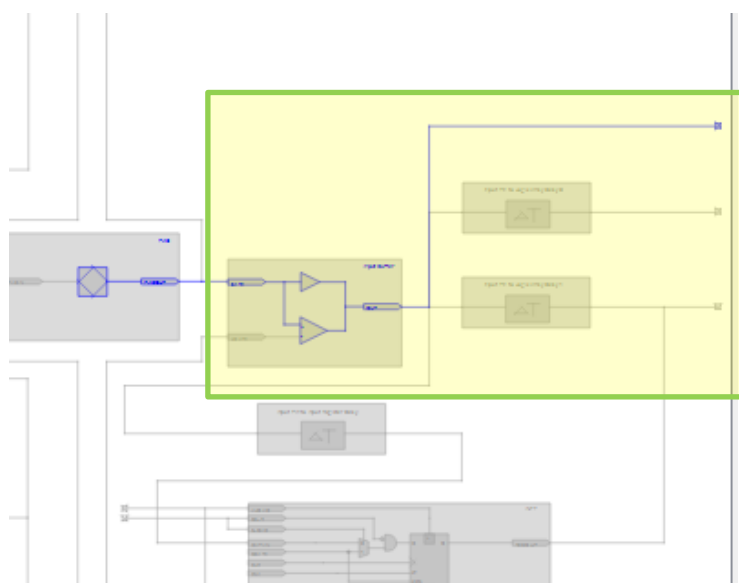


Рис. 80 Просмотр прохождения сигнала SW0 через цепи блока ввода-вывода микросхемы

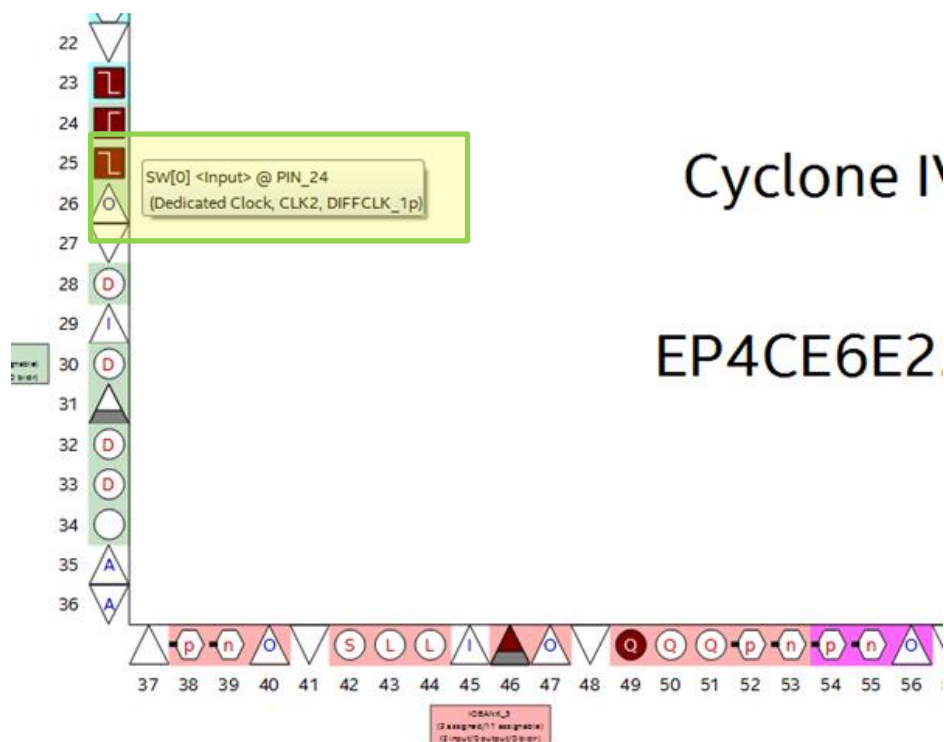



Рис. 81 Просмотр свойств сигнала SW0 около ввода-вывода микросхемы

Выводы, к которым подключены сигналы **SW[1]**, **SW[7..4]**, относятся к той же группе выделенных выводов. Проверьте эту информацию для сигнала **SW[5]**, сохраните скриншоты для отчета.

14.10. В окне **Pin Planner** для сигнала **SW[2]** через перекрестные ссылки откройте окно **Chip Planner**.

14.11. Выберите **PLL** в нижнем левом углу (он используется в проекте). Затем

последовательно нажмите инструмент  на левой панели инструментов 3 раза (рис. 82). Теперь видно, что тактовые сигналы поступают в блоки ввода и вывода микросхемы, т.е. компилятор задействовал имеющиеся там триггеры согласно назначениям (за исключением сигналов, которые были обсуждены выше). Сохраните скриншот для отчета.

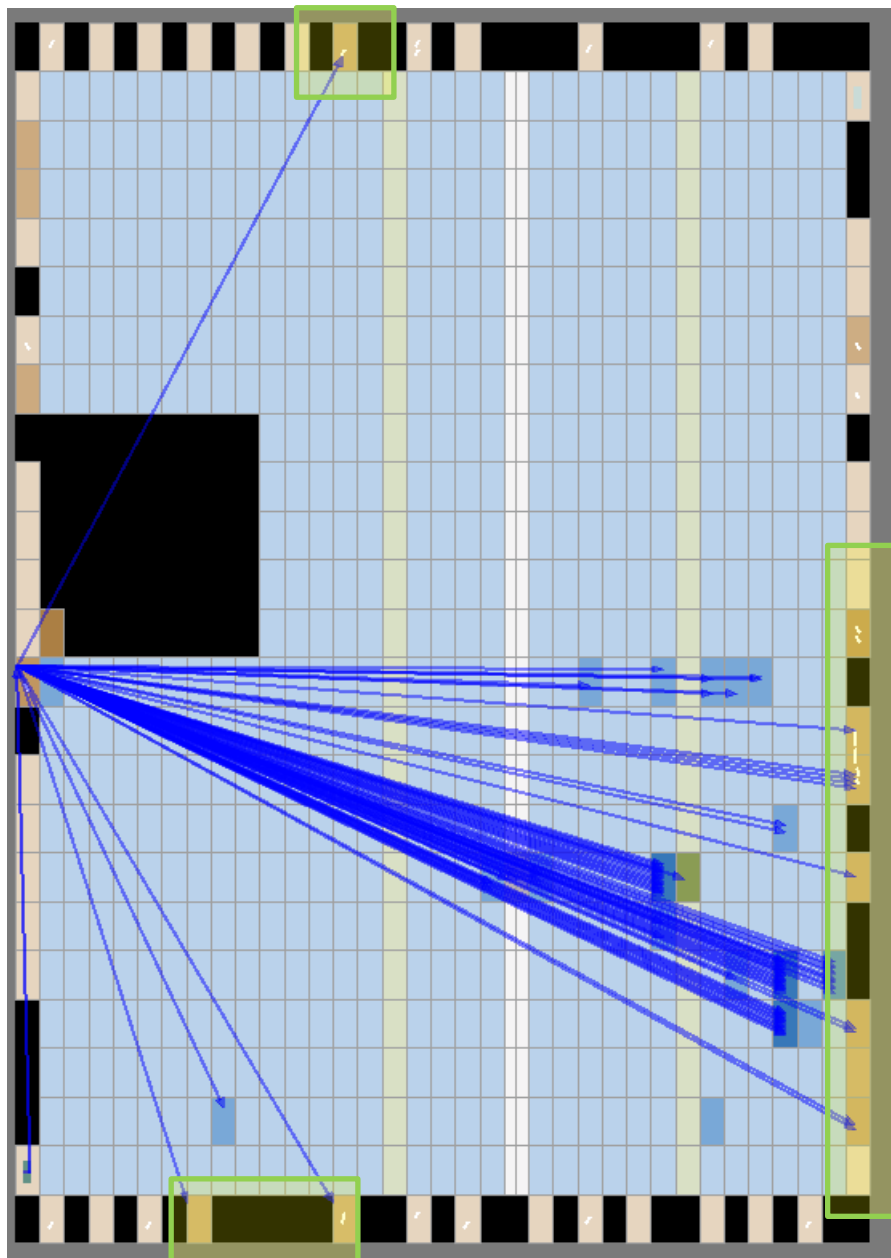

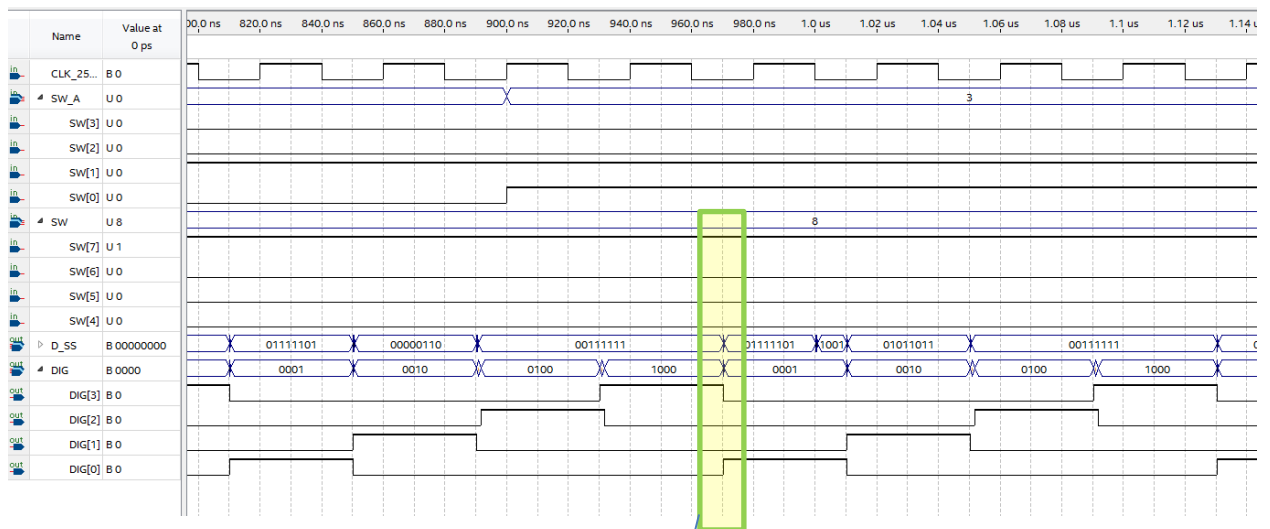


Рис. 82 Просмотр прохождения сигналов тактирования внутри кристалла

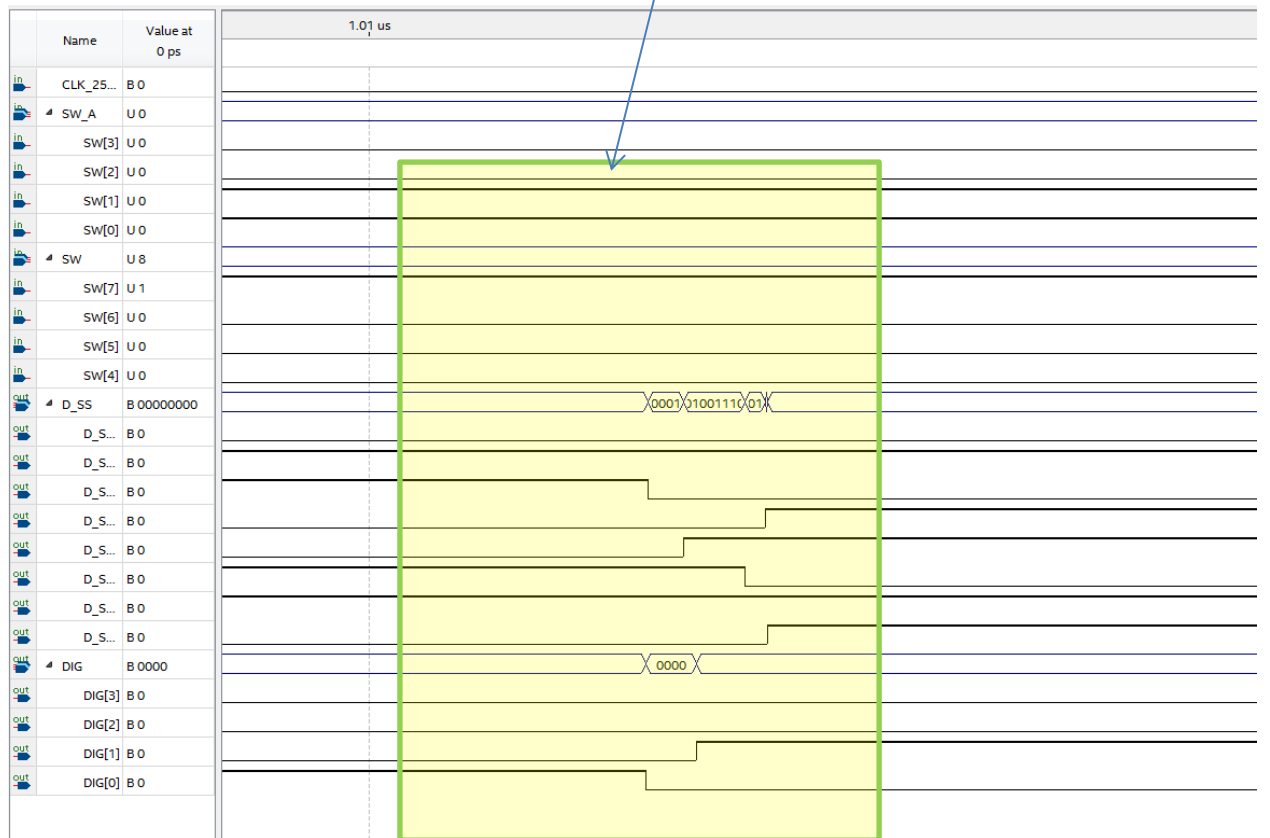
## 15. Временное моделирование проекта

15.1. Откройте файл **lab4\_iav.vwf**.

15.2. Запустите симуляцию нажатием на кнопку  (**Run Timing Simulation**) или выполните **Simulation -> Run Timing Simulation**. На эпюрах с результатами симуляции (рис. 83) видны переходные процессы на выходных сигналах, связанные с разным временем формирования результатов на разрядах сигналов **DIG[3..0]** и **D\_SS[7..0]**.



a)



b)

Рис.83 Результат временной симуляции схемы проекта в исходном масштабе (a) и в увеличенном масштабе (b)

Если в результате симуляции появится окно **Simulation Flow Progress** с отображением ошибки, выполните **Simulation -> Simulation Settings ->** Выберите вкладку **Timing Simulation Setting -> Restore Defaults-> Save**. Затем повторите временную симуляцию.

## 16. Изменение параметров схемы проекта и его повторная полная компиляция

16.1. Для возможности симуляции за разумное время, формирование развертки изображения на индикаторе производится в ускоренном режиме. Это обеспечивается генерацией более высокой частоты сигнала **ENA**, по которому происходит смена состояний автомата по формированию сигналов развертки для индикатора. Эта частота, в свою очередь, формируется счетчиком **Counter\_12b** и зависит от его модуля счета. Для симуляции модуль установлен в значение **4**. Для реальной работы схемы на отладочной плате необходимо частоту сигнала **ENA** снизить, иначе элементы схемы развертки индикатора не будут успевать отрабатывать необходимый алгоритм смены сигналов (не хватит быстродействия переключения транзисторов, например). Поэтому необходимо установить модуль счетчика, формирующего сигнал **ENA**, в большее значение. Для данной схемы приемлимое значение равно **4095**. Модуль счетчика меняется с помощью **MegaWizard Plug-In Manager**.

16.2. Откройте файл со схемой проекта **lab4\_iav.bdf** и дважды кликните ЛКМ на компонент **ind\_drv**. В появившемся окошке **Select File** выберите файл **ind\_drv.bdf** (рис. 84), далее «**Ok**», откроется окно схематического редактора со схемой компонента **ind\_drv**.

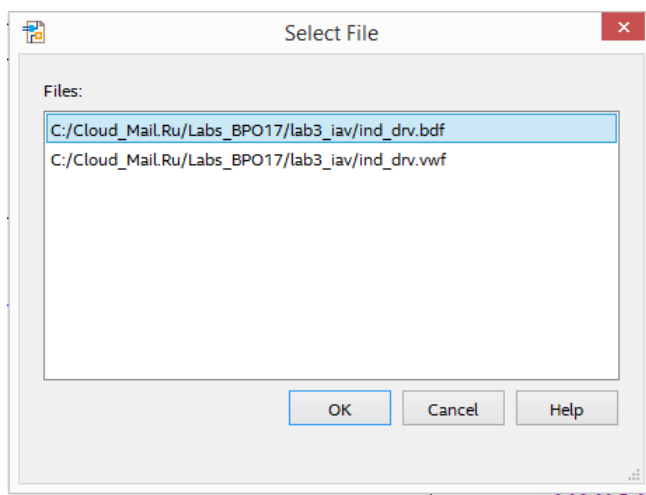


Рис. 84 Выбор файла для изменений

16.3. Дважды кликните на компонент **Counter\_12b**, откроется **MegaWizard Plug-In Manager**. На странице **2** измените значение **modulus** с **4** на **4095** (рис. 85), далее на странице **5** выполните **Finish**. Далее в окне **MegaWizard Plug-In Manager** (рис. 86) выполните «**Ok**», подтвердите **Update** символа **Counter\_12b** (рис. 87а) и в следующем окне **Update Symbol or Block** выполните «**Ok**» (рис. 87б). На символе компонента появится новое значение модуля: **4095** (рис. 88).

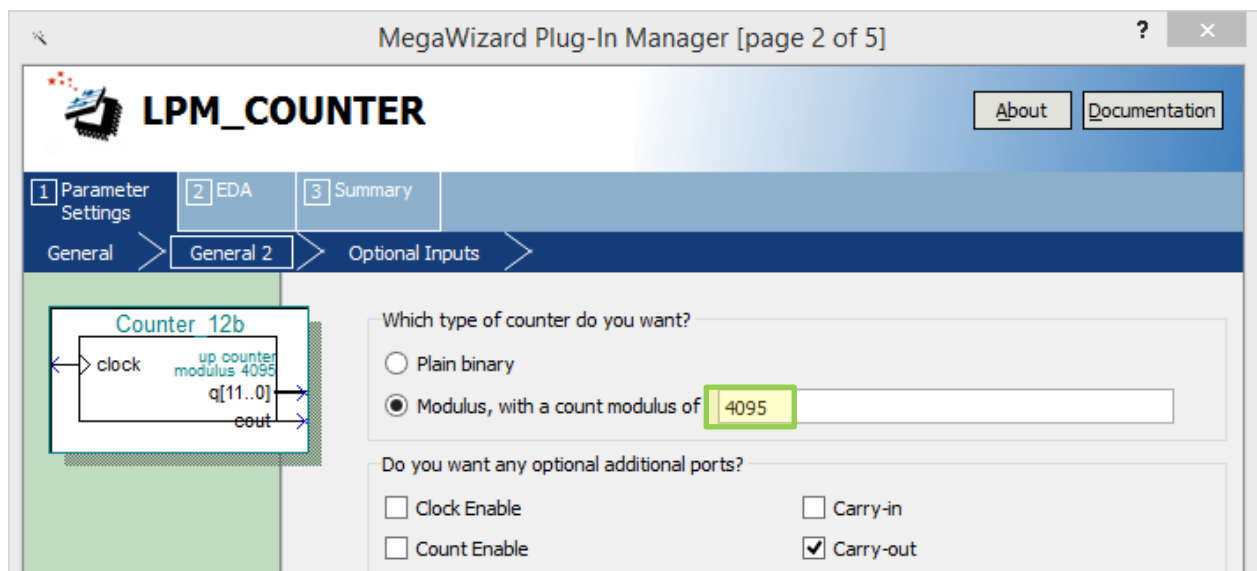


Рис. 85 Изменение модуля счетчика

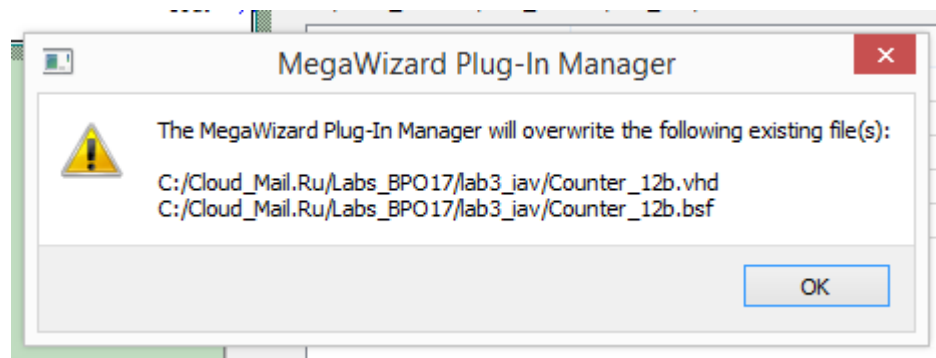
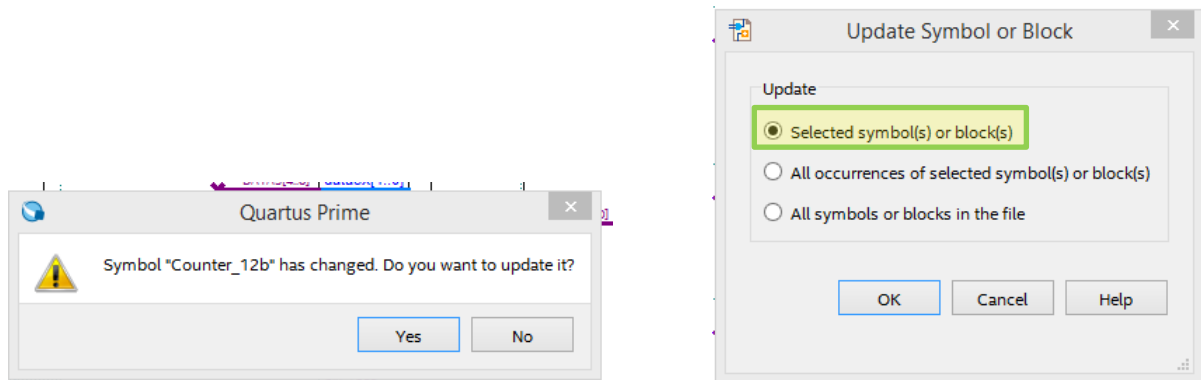


Рис. 86 Обновление компонента Counter\_12b



a

b

Рис. 87 Обновление символа Counter\_12b в схеме



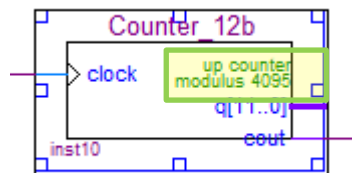


Рис. 88 Обновление символа Counter\_12b в схеме

16.4. Сохраните схему и выполните полную компиляцию.

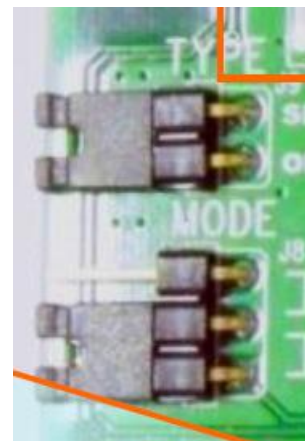
## 17. Конфигурирование ПЛИС и проверка работоспособности на отладочной плате.

17.1. Подсоедините к отладочной плате модуль 4-х разрядного 7-сегментного светодиодного индикатора.

17.2. Подключите отладочную плату **miniDLab-CIV** к компьютеру с помощью **USB** кабеля (тип A–miniB), и включите питание (рис. 89а). Проверьте правильность установки перемычек (рис. 89b).




а



б

Рис. 89 Переключатель Power(b) и конфигурация разъемов установки режимов работы платы (а)

17.3. Откройте окно программатора **Programmer** из окна **Tasks (Program Device)** или кнопкой  или **Tools->Programmer** (рис. 90).

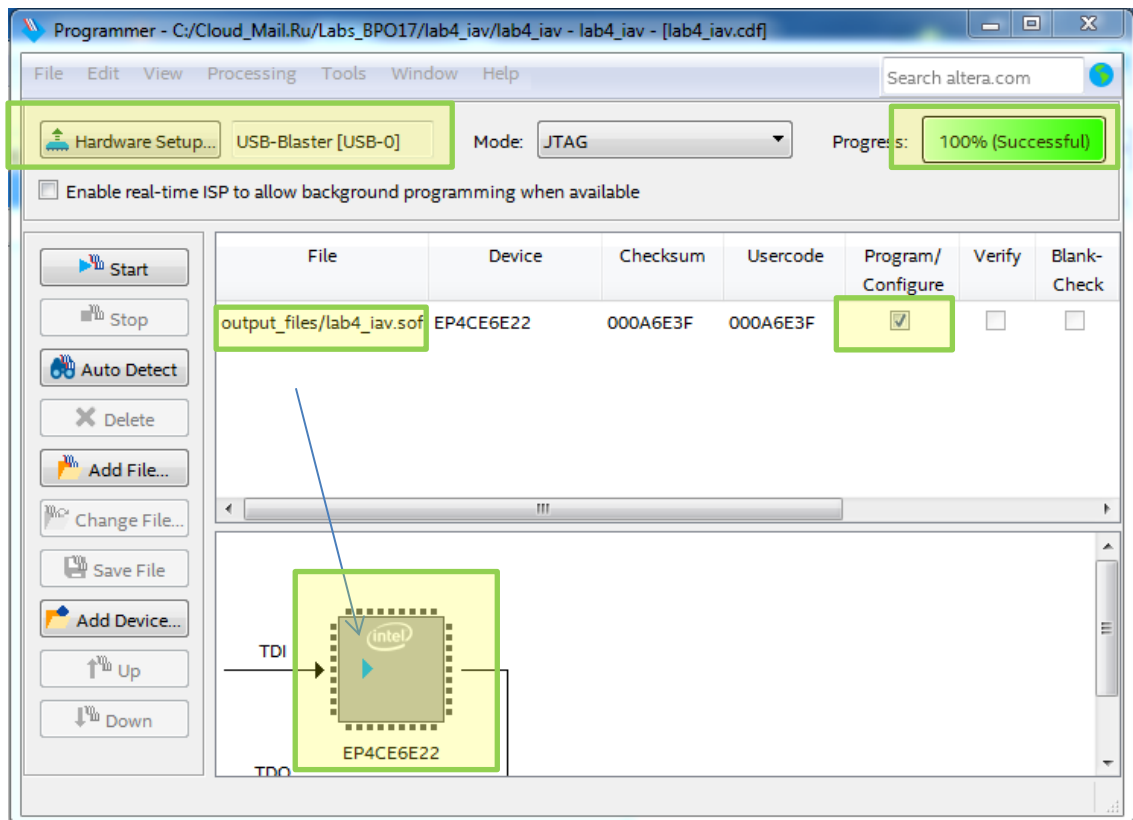


Рис.90 Окно программатора

17.4. Если **USB blaster** в программаторе не виден, выберите его через кнопку **Hardware Setup**.

17.5. Если конфигурационный файл автоматически не загрузился, выберите его с помощью кнопки **Add File** из папки **output\_files** проекта, он хранится в файле с расширением **sof** (**SRAM Object File**) **lab4\_iav.sof**.

17.6. Выполните **Start**, в окошке **Progress** будет виден текущий статус загрузки. Скриншот для отчета.

17.7. После успешной загрузки конфигурации ПЛИС проверьте работу проекта. Схема работает следующим образом:

на 4-м разряде (старший разряд) будет высвечиваться символ '0',

на 3-м разряде будет высвечиваться 3 десятичный разряд числа, полученного в результате умножения чисел, формируемых переключателями групп SW\_A и SW\_B.

на 2-м разряде будет высвечиваться 2 десятичный разряд числа, полученного в результате умножения чисел, формируемых переключателями групп SW\_A и SW\_B.

на 1-м разряде (младший разряд) будет высвечиваться 1 десятичный разряд числа, полученного в результате умножения чисел, формируемых переключателями групп SW\_A и SW\_B.

17.8. Выключите и отсоедините отладочную плату от компьютера.

Лабораторная работа завершена.

Общие требования к отчету:

1. Отрадите в отчете порядок выполнения лабораторной работы и занесите в него результаты выполнения ключевых этапов в виде рисунков (путем копирования необходимой информации с экрана монитора) и текстовых пояснений.
2. В выводах отразите суть выполненной работы и полученные навыки.
3. Проанализируйте полученные данные, если это требуется в задании, и сделайте по ним выводы.