Лабораторная работа 1

Изучение IP-компонентов и редактора создания содержимого памяти

САПР Quartus Prime Lite

Цель лабораторной работы: Знакомство с базовыми возможностями схемного редактора (использование IP–компонентов) и редактора создания содержимого памяти в САПР Quartus Prime. Работа включает следующие этапы:

- Создание проекта
- Создание файла инициализации памяти ROM
- Создание экземпляра модуля памяти ROM на базе библиотеки IP-компонентов
- Создание экземпляров счетчика и декодера на базе библиотеки IP-компонентов
- Создание дизайн - файла с помощью схематического редактора
- Назначение выводов ПЛИС
- Создание sdc – файла
- Выполнение полной компиляции проекта
- Конфигурирование ПЛИС и проверка работоспособности на отладочной плате.

Алгоритм работы проекта: Входная тактовая частота значением 25Mhz поступает на 28-разрядный счетчик, сигналы с выходных 4-х старших разрядов которого поступают на преобразователь двоичного кода в 7-сегментный код, реализованный на модуле памяти ROM. Сигналы с ROM поступают на сегменты светодиодного 7-сегментного 4-х разрядного индикатора. Выбор разряда индикатора, который должен отображать текущую цифру, определяемую текущим кодом с выхода счетчика, осуществляется с помощью декодера. Декодер преобразует состояние 2-х ползунковых переключателей SW5, SW4 в 4 сигнала, каждый из которых разрешает работу соответствующего разряда индикатора. Одновременно может отображаться информация только на одном разряде индикатора.

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 **lab1_<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 файлов.*

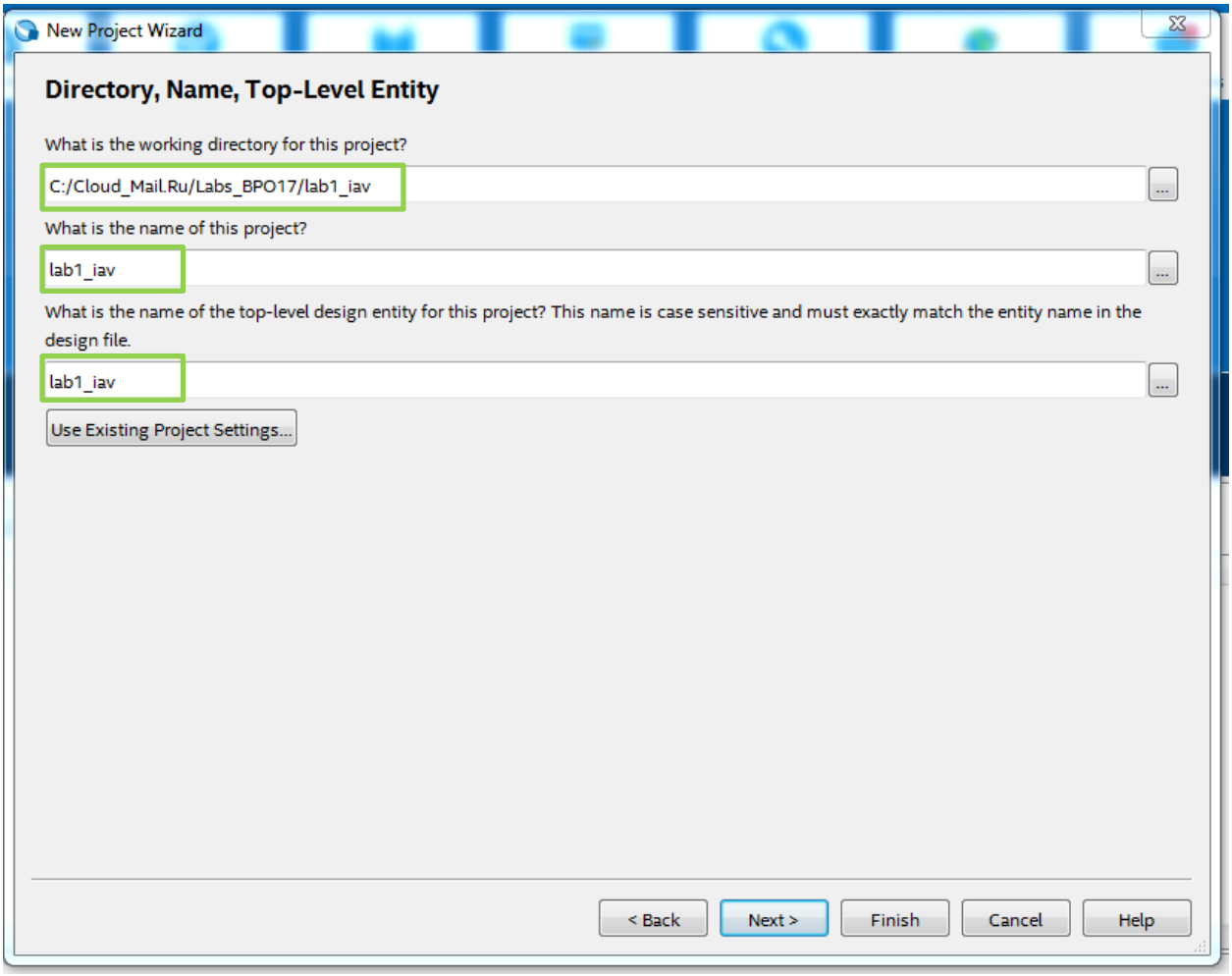


Рис.1 Окно выбора директории для установки проекта

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**».

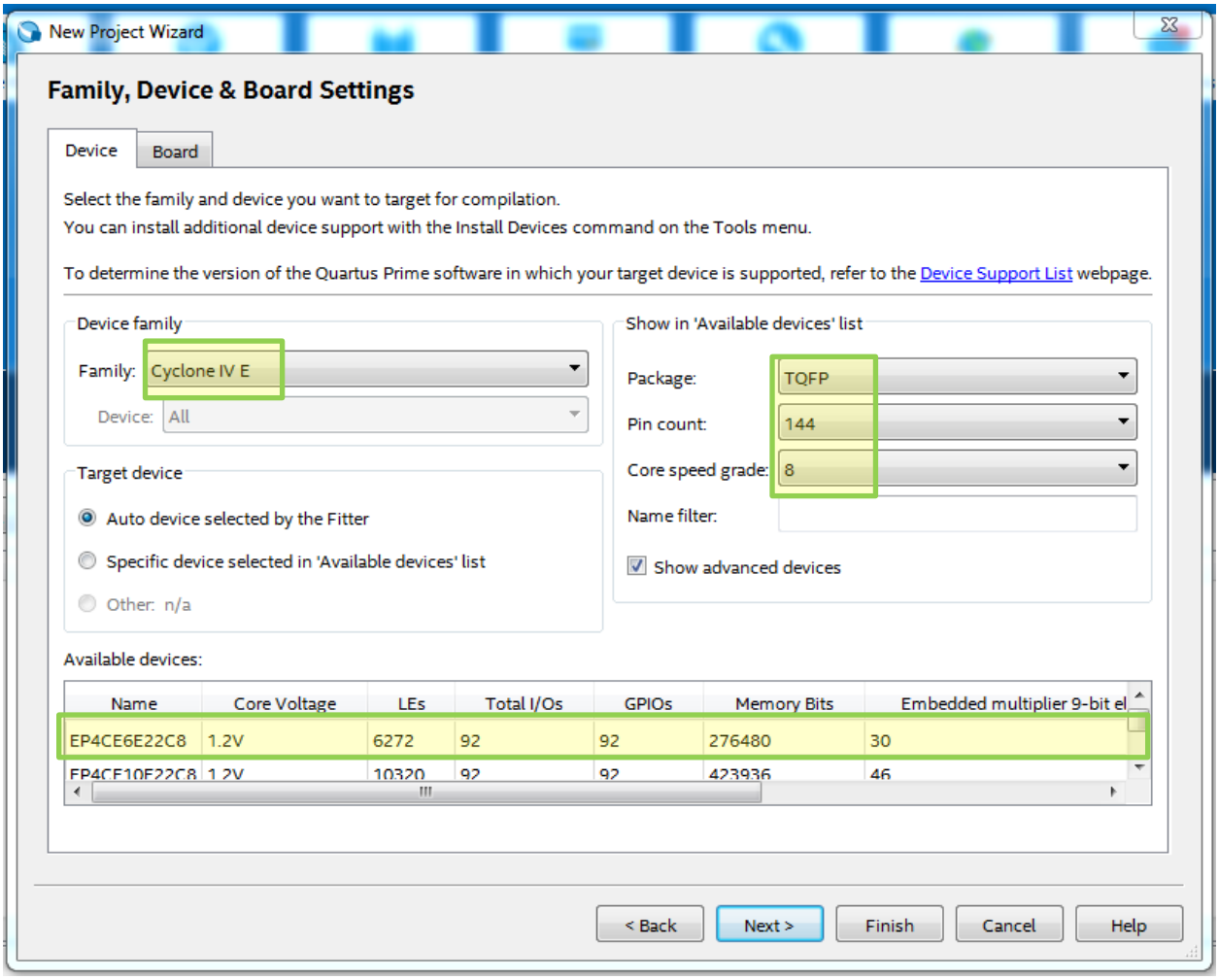


Рис.2 Окно для выбора микросхемы

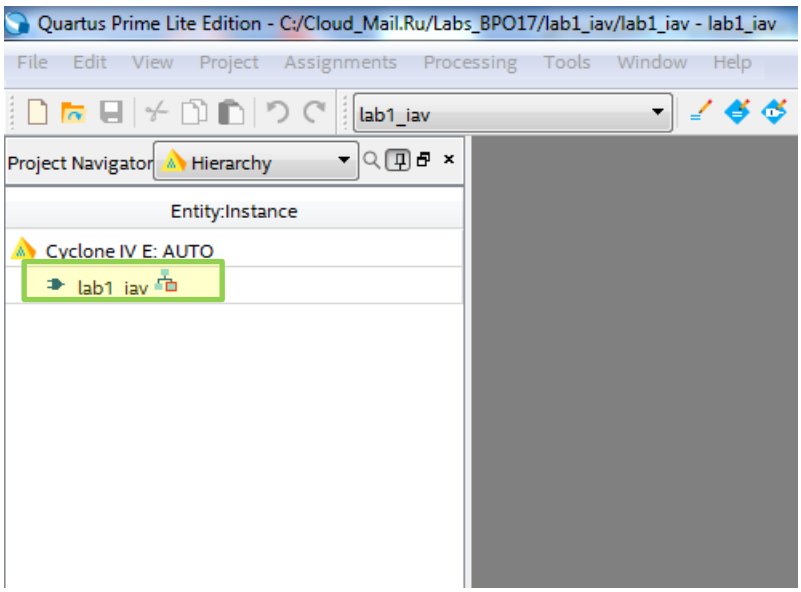


Рис. 3 Окно пакета с созданным проектом

После этого вновь созданный проект становится текущим в запущенной системе проектирования **Quartus Prime** (рис. 3).

2. Создание файла инициализации памяти ROM

2.1. Выполните «**File -> New**», откроется окошко**"New"** (рис.4) с выбором типов создаваемых файлов.

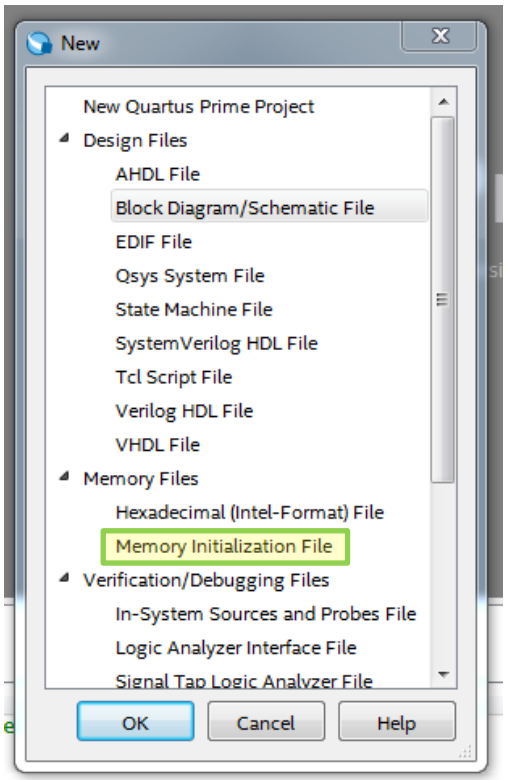


Рис. 4 Окно выбора типа создаваемого проектного файла (design file)

2.2. Выберите **Memory Files -> *Memory Initialization File***, далее «**OK**».

2.3. В появившемся окошке **“Number of Words &Word Size”** задайте параметры памяти ROM, для которой создается файл инициализации (рис. 5). Далее «**OK**». Откроется окно редактора памяти (рис.6)

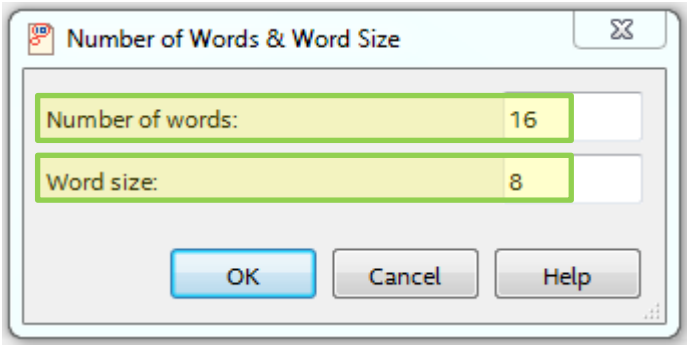


Рис. 5 Окно выбора параметров памяти

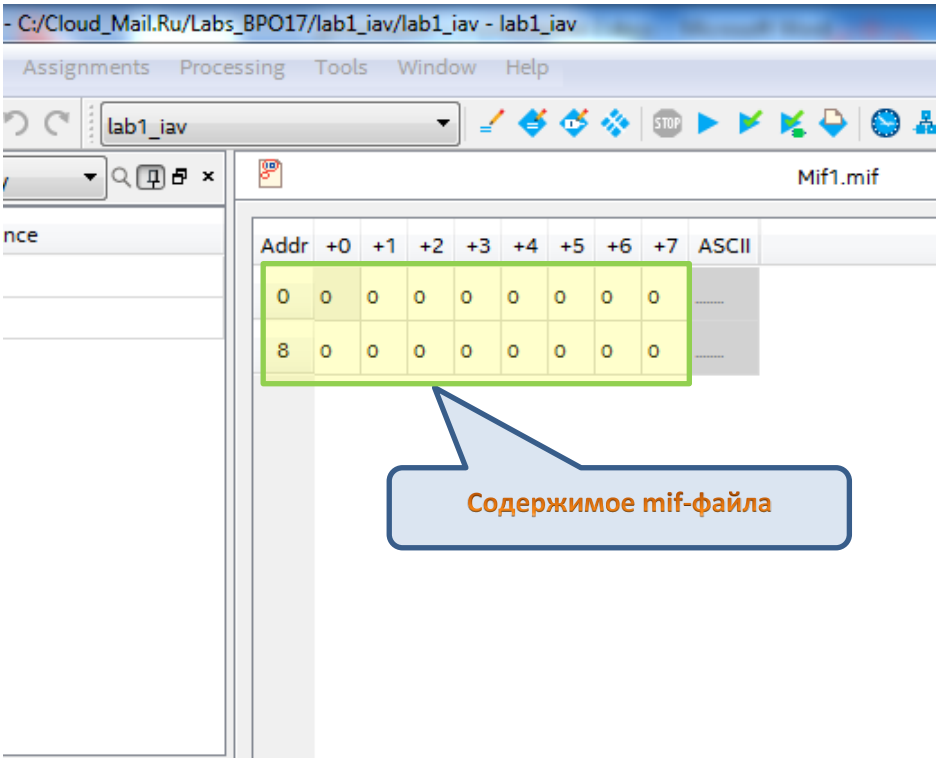
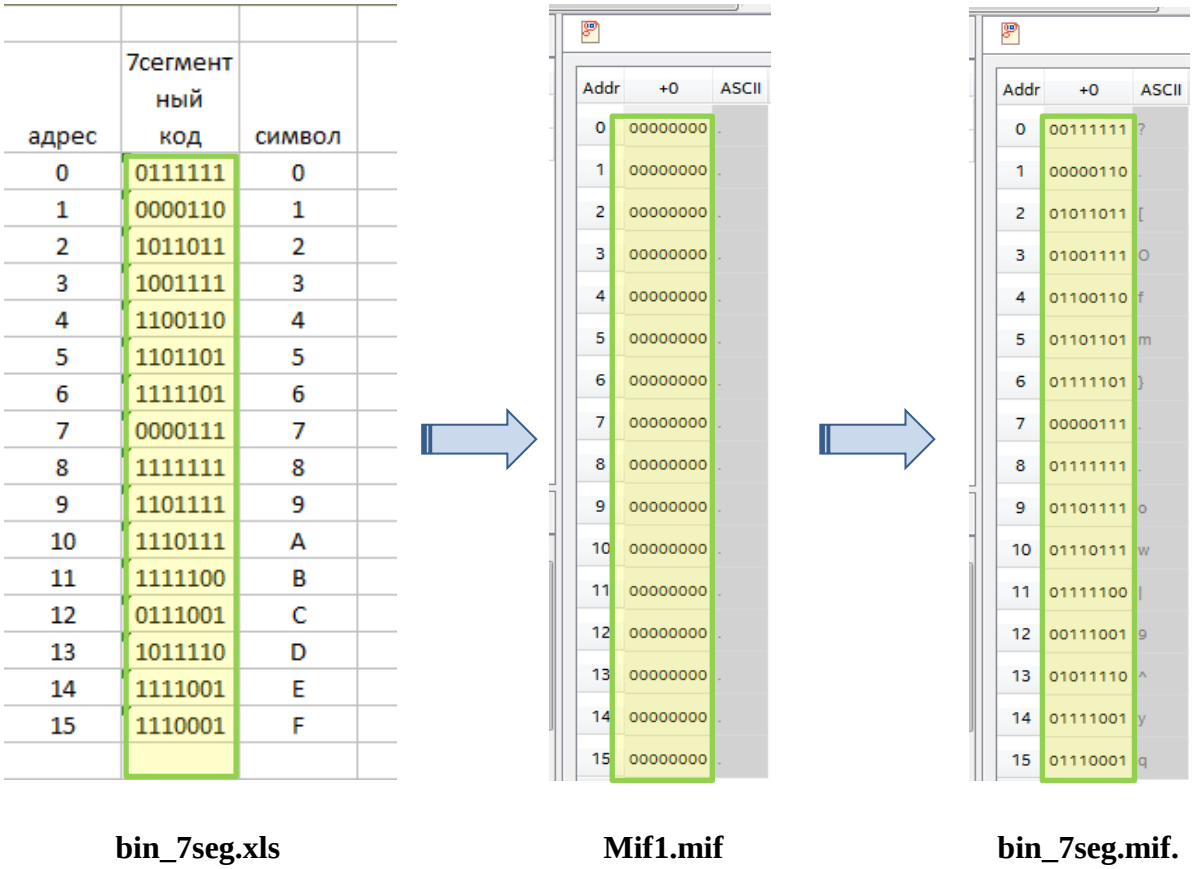


Рис. 6 Окно редактора памяти

2.4. Выполните **«View->Memory Radix->Binary»**, затем **«View->Cells Per Row->1».** Окно редактора памяти видоизменится следующим образом (рис. 7, **Mif1.mif**):



адрес	7сегментный код	символ
0	0111111	0
1	0000110	1
2	1011011	2
3	1001111	3
4	1100110	4
5	1101101	5
6	1111101	6
7	0000111	7
8	1111111	8
9	1101111	9
10	1110111	A
11	1111100	B
12	0111001	C
13	1011110	D
14	1111001	E
15	1110001	F

Addr	+0	ASCII
0	00000000	.
1	00000000	.
2	00000000	.
3	00000000	.
4	00000000	.
5	00000000	.
6	00000000	.
7	00000000	.
8	00000000	.
9	00000000	.
10	00000000	.
11	00000000	.
12	00000000	.
13	00000000	.
14	00000000	.
15	00000000	.

Addr	+0	ASCII
0	00111111	?
1	00000110	.
2	01011011	[
3	01001111	O
4	01100110	f
5	01101101	m
6	01111101	}
7	00000111	.
8	01111111	.
9	01101111	o
10	01110111	w
11	01111100	|
12	00111001	9
13	01011110	^
14	01111001	y
15	01110001	q

bin_7seg.xls **Mif1.mif** **bin_7seg.mif.**

Рис. 7 Окно редактора памяти

2.5. Откройте файл **bin_7seg.xls** и скопируйте содержимое колонки «7сегментный код» в созданный **mif** – файл (рис. 7). Сохраните под именем **bin_7seg.mif.**

3. Создание экземпляра модуля памяти ROM

3.1. В разделе **Library** окна **IP Catalog** выберите мегафункцию **ROM: 1-PORT** (рис. 8) и нажмите кнопку «**+ADD**»,

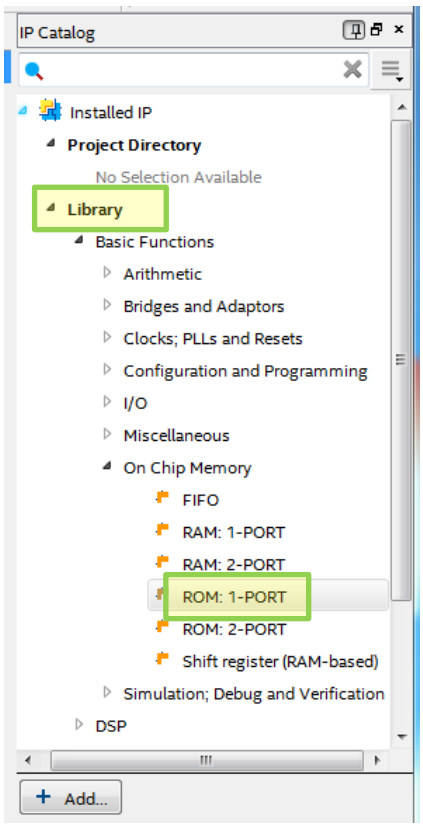


Рис. 8 Окно **IP Catalog**

3.2. 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.9). Выполните «**OK**».

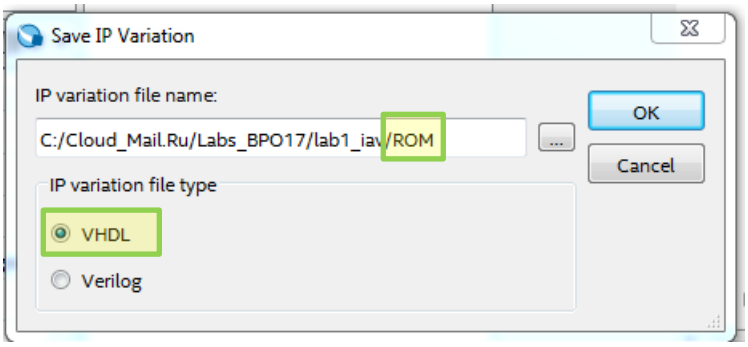


Рис. 9 Окно создания экземпляра выбранного компонента

3.3. В появившемся окне **“MegaWizared Plug-In Manager [page 1 of 5]”** установите параметры как на рис. 10. Выполните «**Next».**

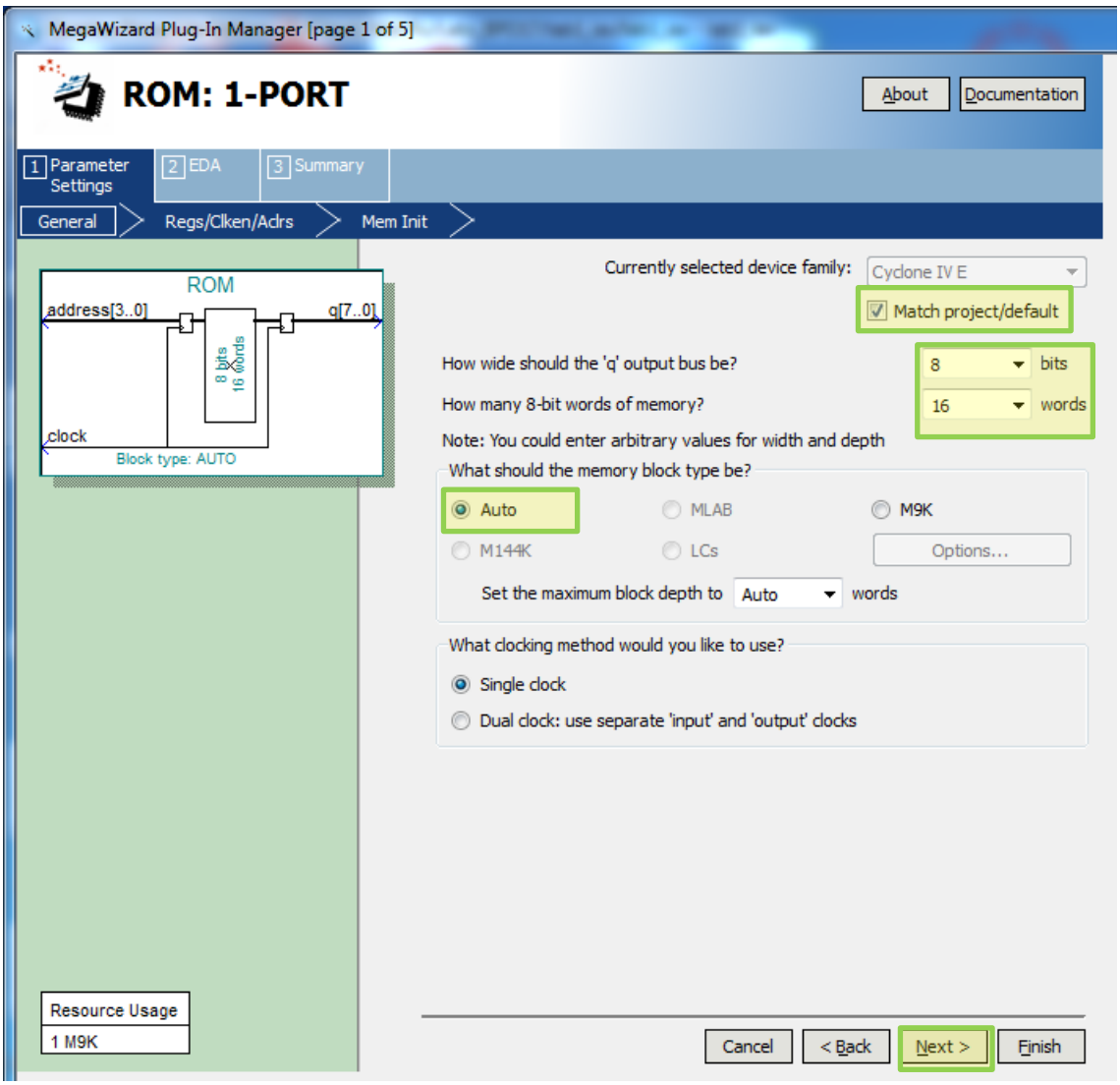


Рис. 10 Окно настройки экземпляра выбранного компонента (page 1)

3.4. В появившемся окне **“MegaWizared Plug-In Manager [page 2 of 5]”** параметры должны быть такие же, как на рис. 11. Выполните «**Next».**

3.5. В появившемся окне **“MegaWizared Plug-In Manager [page 3 of 5]”** установите параметры как на рис. 12. В окошке **File name** с помощью кнопки **Browse** выберите созданный ранее файл **bin_7seg.mif** (рис. 13). Выполните «**Open»,** затем «**Next».**

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

3.7. В появившемся окне **“MegaWizared 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_7seg.mif.** Файл **ROM.vhd** содержит информацию о текстовом описании созданного экземпляра компонента (vhd – сокращение от названия зыка 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.

3.8. В появившемся окне **“QUARTUS Prime IP Files**”, на предложение добавить созданный компонент в проект (рис. 15), выберите «**Yes».**

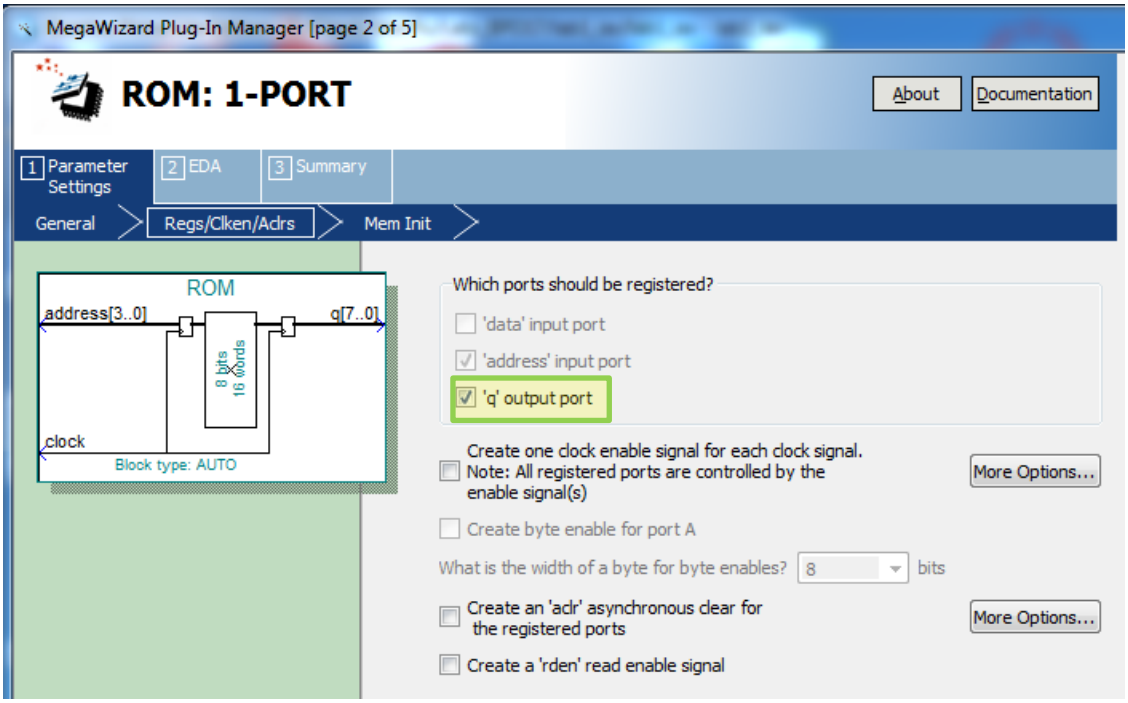


Рис. 11 Окно настройки экземпляра выбранного компонента (page 2)

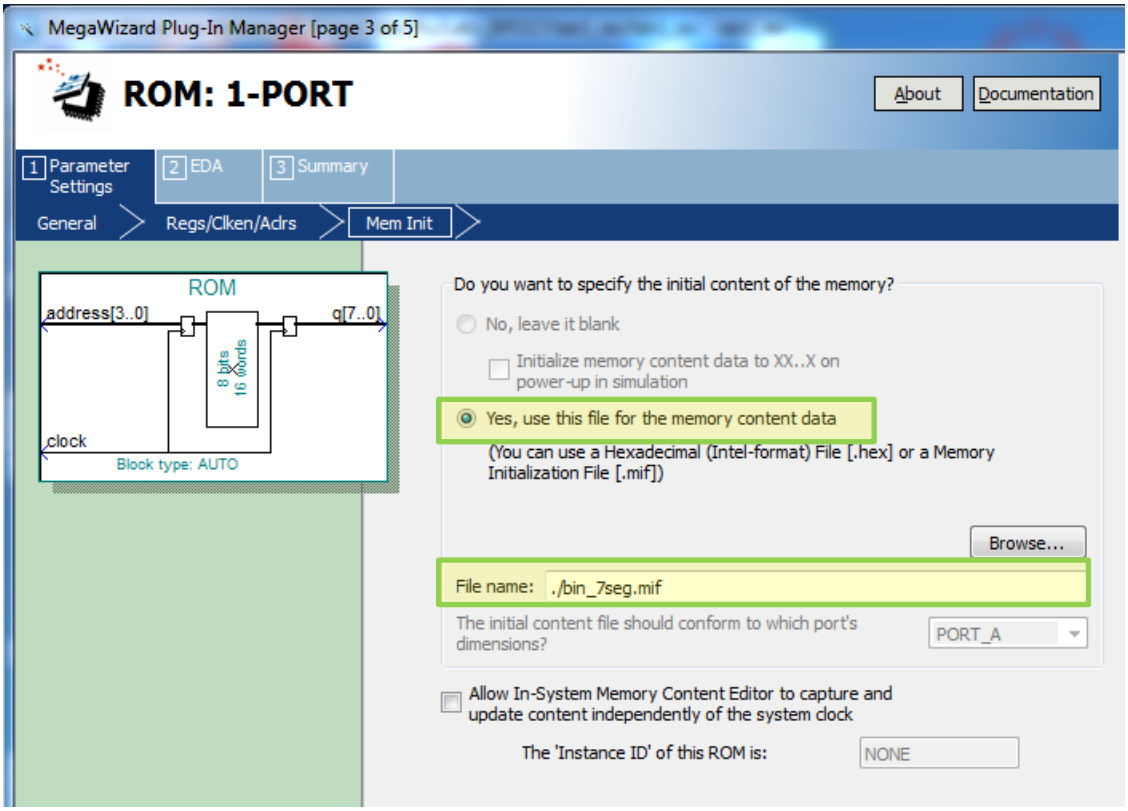


Рис. 12 Окно настройки экземпляра выбранного компонента (page 3)

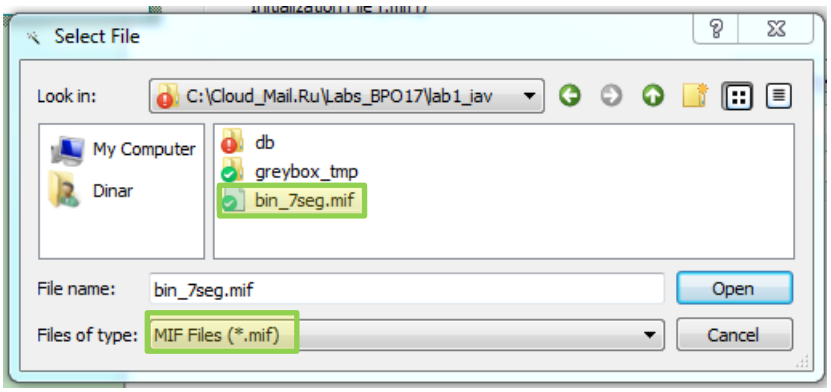


Рис. 13 Окно выбора файла инициализации памяти ROM

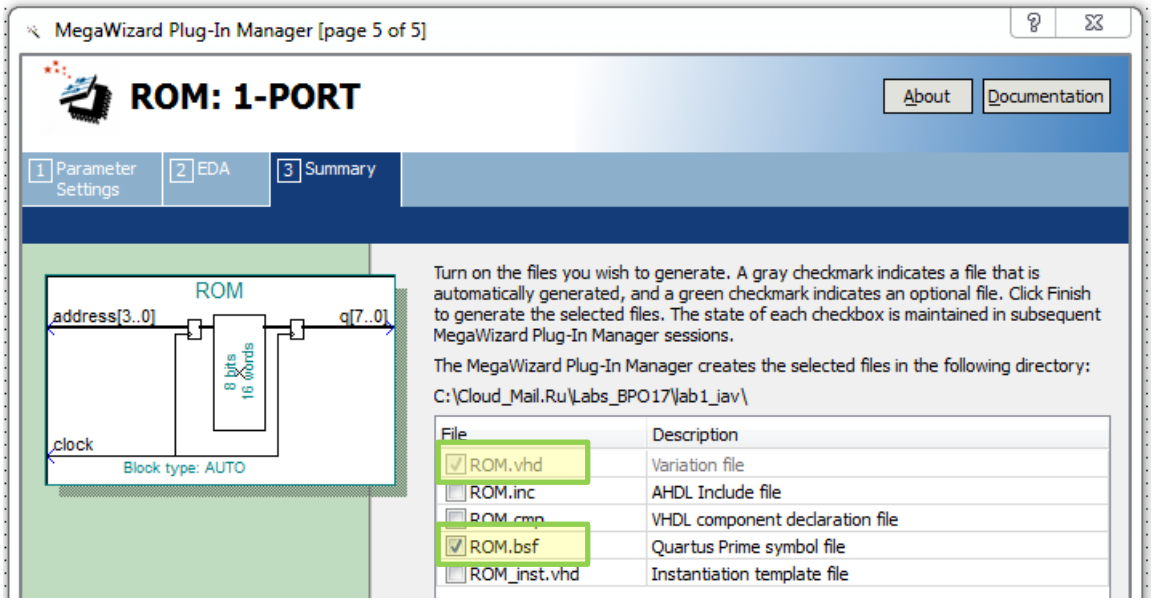


Рис. 14 Окно настройки экземпляра выбранного компонента (page 5)

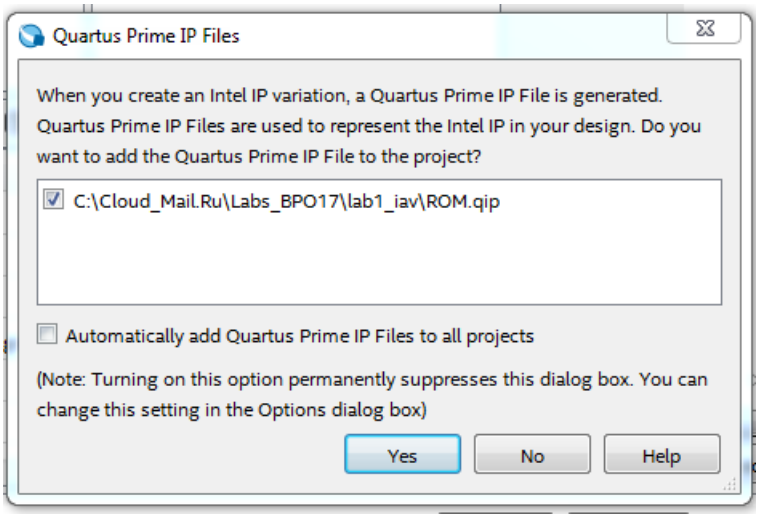


Рис. 15 Окно подключения созданного экземпляра компонента в проект

4. Создание экземпляров счетчика и декодера на базе библиотеки IP-компонентов

4.1. В окне **IP Catalog** в разделе **Library->Basic Functions->Arithmetic** выберите мегафункцию **LPM_COUNTER** (рис. 16) и нажмите кнопку «**+ADD**»,

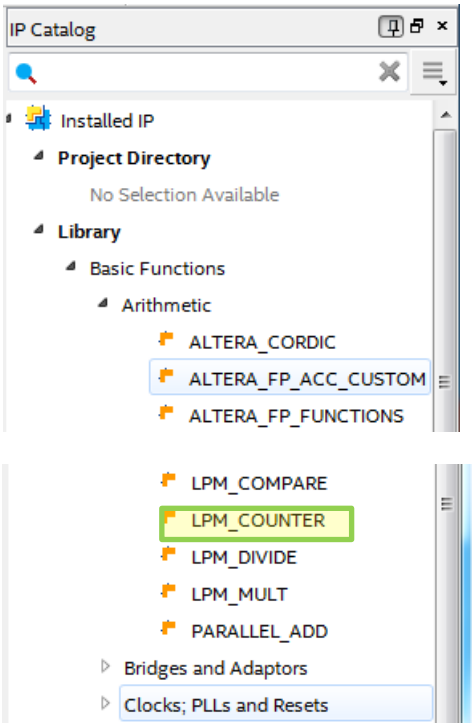


Рис. 16 Окно **IP Catalog**

4.2. В появившемся окне настройки задайте имя экземпляра выбранного компонента (Counter_28b) и укажите язык описания аппаратуры (VHDL), в коде которого он будет храниться в проекте (рис.17). Выполните «**OK**».

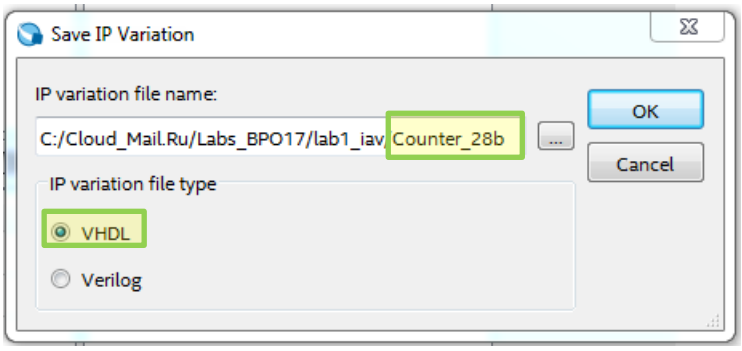


Рис. 17 Окно создания экземпляра выбранного компонента

4.3. В появившемся окне **“MegaWizared Plug-In Manager [page 1 of 5]”** установите разрядность счетчика 28 бит (рис. 18). Выполните «**Next».**

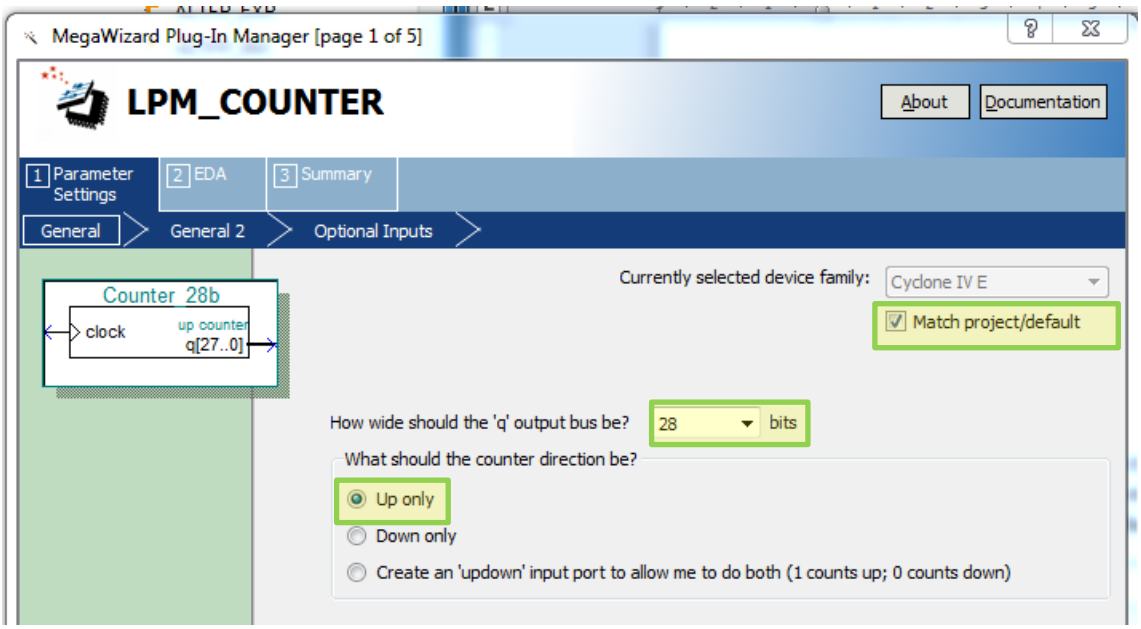


Рис. 18 Окно настройки экземпляра выбранного компонента (page 1)

4.4. В появившемся окне **“MegaWizared Plug-In Manager [page 2 of 5]”** выполните «**Next».**

4.5. В появившемся окне **“MegaWizared Plug-In Manager [page 3 of 5]”** выполните «**Next».**

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

4.7. В появившемся окне **“MegaWizared Plug-In Manager [page 5 of 5**]” выберите создаваемые файлы, как на рис. 19. Далее «**Finish».** Экземпляр счетчика **Counter_28b** создан.

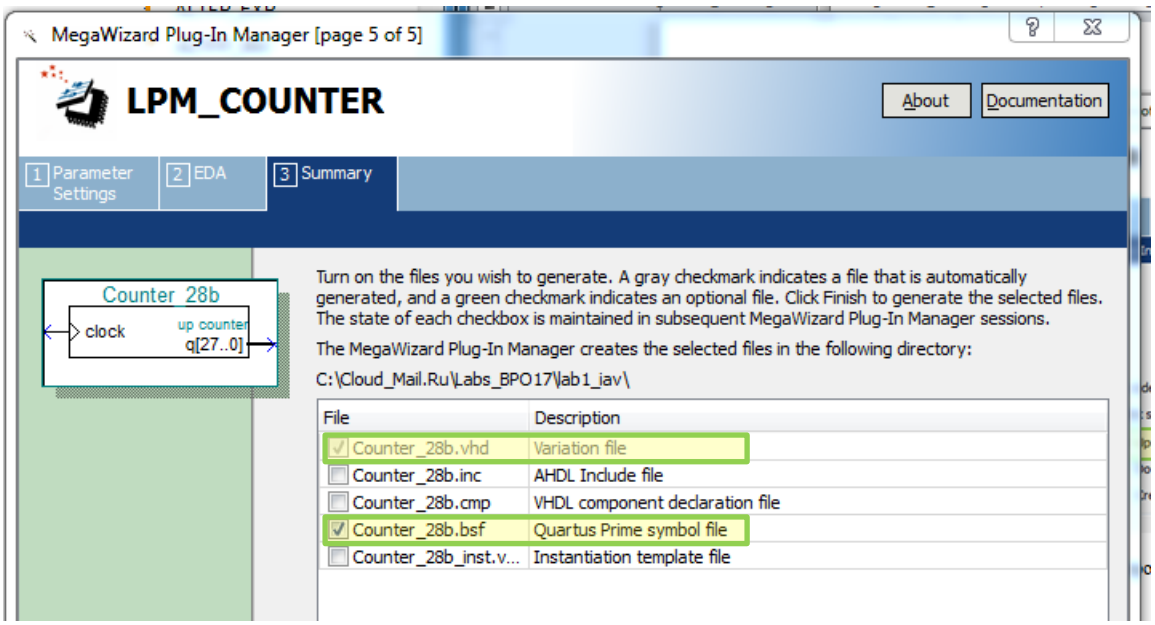


Рис. 19 Окно настройки экземпляра выбранного компонента (page 5)

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

4.9. В окне **IP Catalog** в разделе **Library->Basic Functions->Miscellaneous** выберите мегафункцию **LPM_DECODER** (рис. 20) и нажмите кнопку «**+ADD**»,

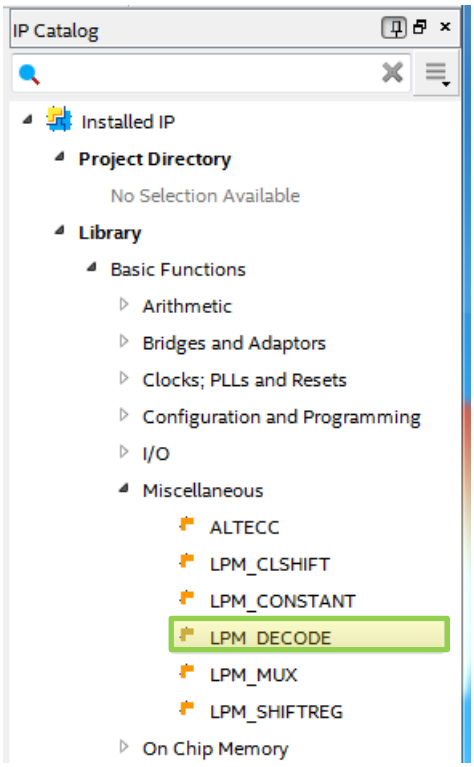


Рис. 20 Окно **IP Catalog**

4.10. В появившемся окне настройки задайте имя экземпляра выбранного компонента (Decoder_2to4) и укажите язык описания аппаратуры (VHDL), в коде которого он будет храниться в проекте (рис.21). Выполните «**OK**».

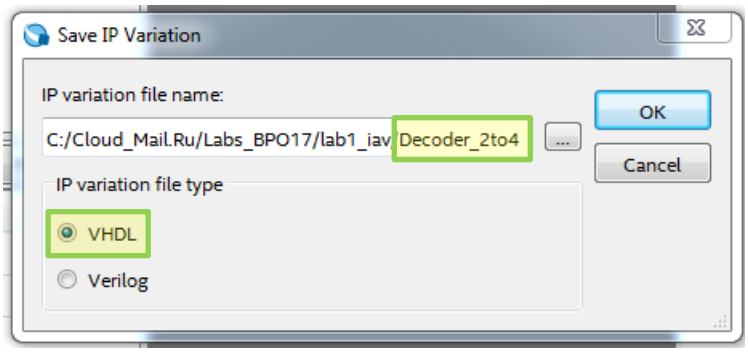


Рис. 21 Окно создания экземпляра выбранного компонента

4.11. В появившемся окне **“MegaWizared Plug-In Manager [page 1 of 4]”** установите разрядность входа 2 бита (рис. 22), нажмите кнопку **“<<Add all”.** Выполните «**Next».**

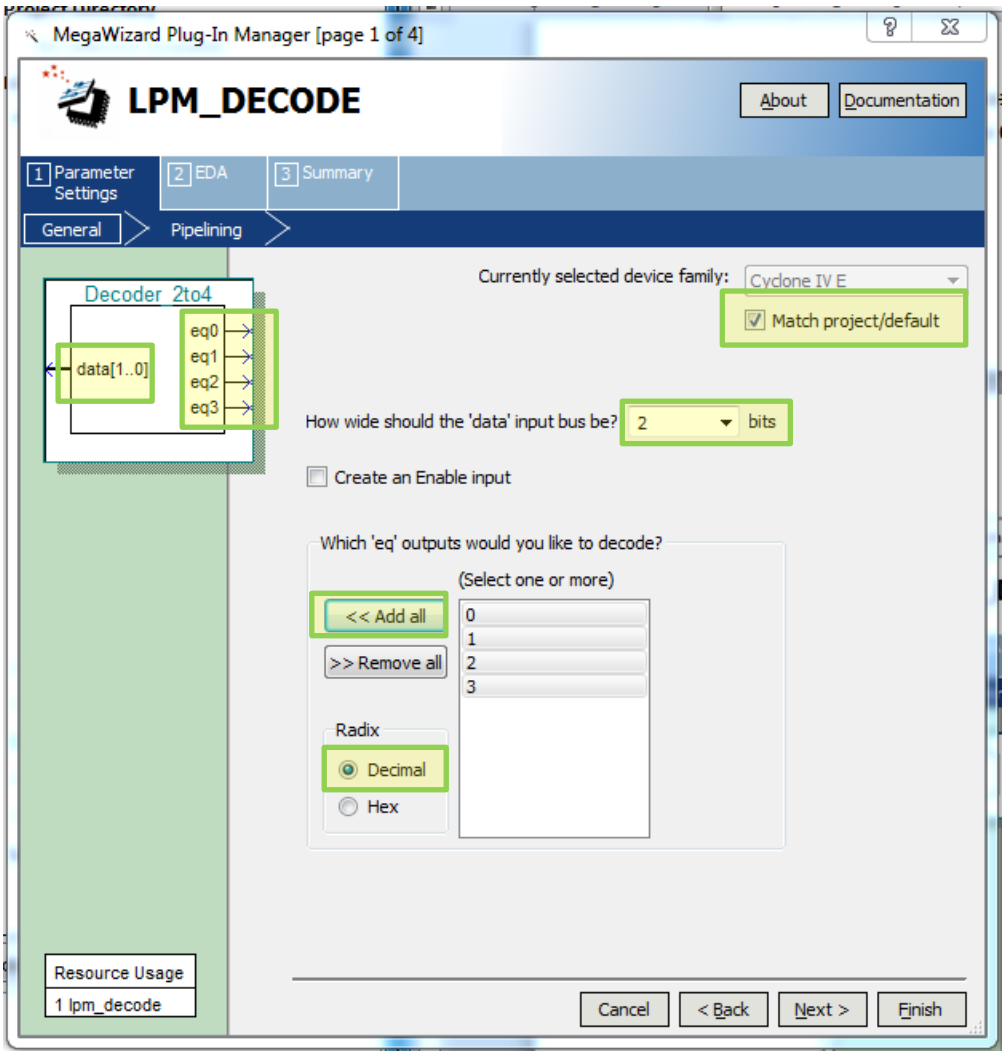


Рис. 22 Окно настройки экземпляра выбранного компонента (page 1)

4.12. В появившемся окне **“MegaWizared Plug-In Manager [page 2 of 4]”** выполните «**Next».**

4.13. В появившемся окне **“MegaWizared Plug-In Manager [page 3 of 4]”** выполните «**Next».**

4.14. В появившемся окне **“MegaWizared Plug-In Manager [page 4 of 4**]” выберите создаваемые файлы (рис. 23). Далее «**Finish».** Экземпляр счетчика **Decoder_2to4** создан.

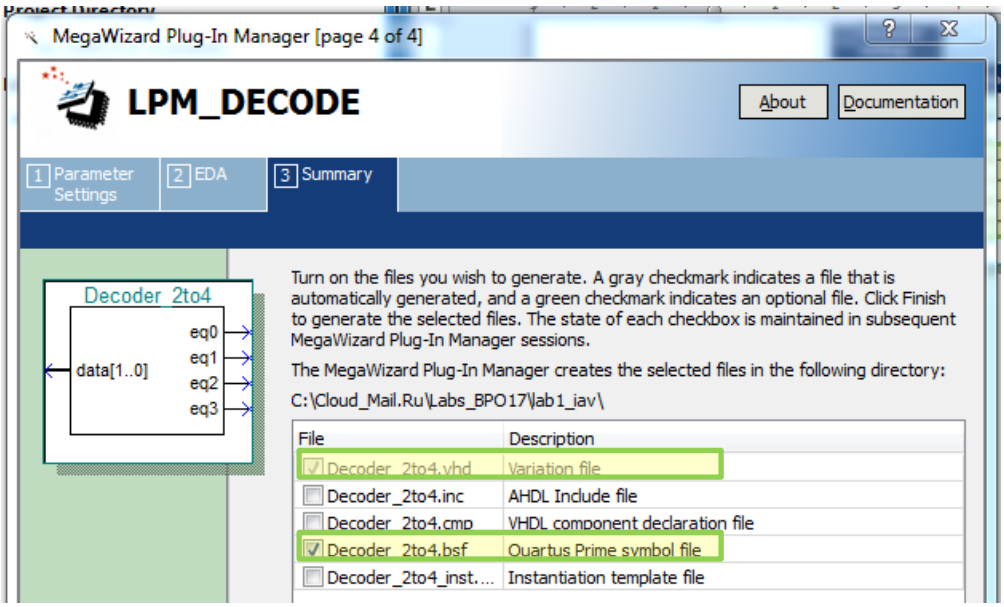
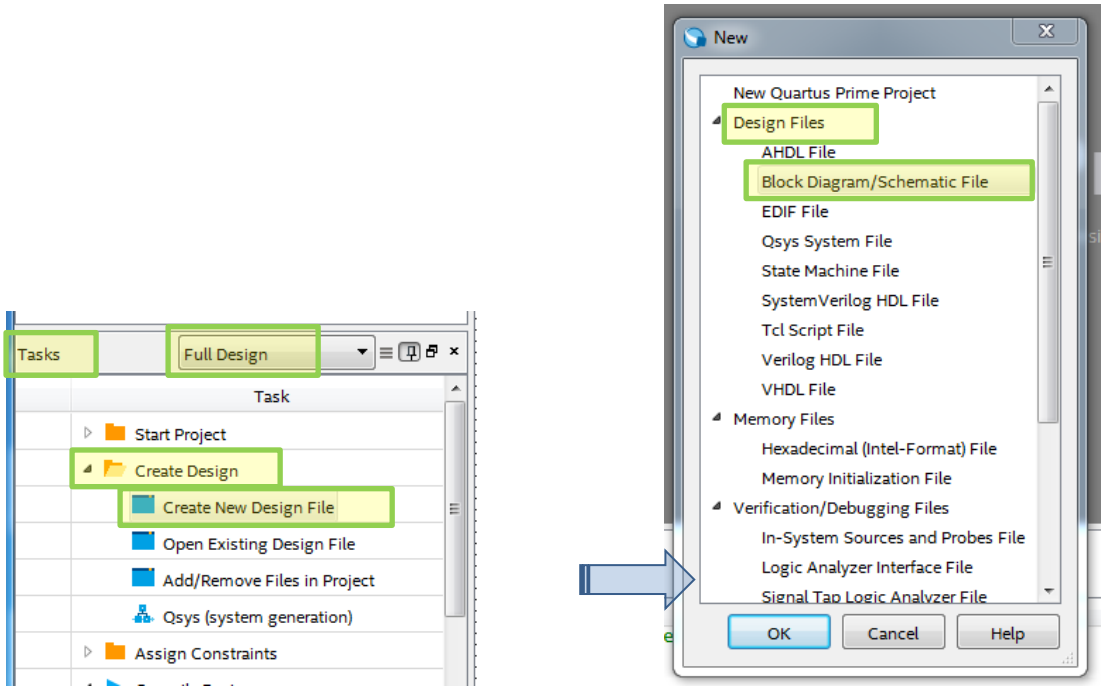


Рис. 23 Окно настройки экземпляра выбранного компонента (page 5)

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

5. Создание дизайн - файла с помощью схематического редактора

5.1. Выполните **File -> New** или **Create New Design** в окне **Task** (рис. 24a), откроется окошко (рис.24b) с выбором типов создаваемых файлов.



a b

Рис. 24 Окно создания проектного файла (a) и окно выбора типа создаваемого проектного файла(b)

Выберите ***Block Diagram/Schematic File*** , далее «**OK**». В результате откроется окошко схематического редактора (**Schematic editor**).

5.2. Сохраните файл со схемой **File -> Save As** с именем ***lab1_iav.bdf.*** Пакет по умолчанию предлагает имя файла с именем проекта и расширением, которое определяется выбранным типом проектного файла (рис. 25). По умолчанию опция ‘**Add file to current project**’ включена и при сохранении созданный файл автоматически подключится к текущему проекту.

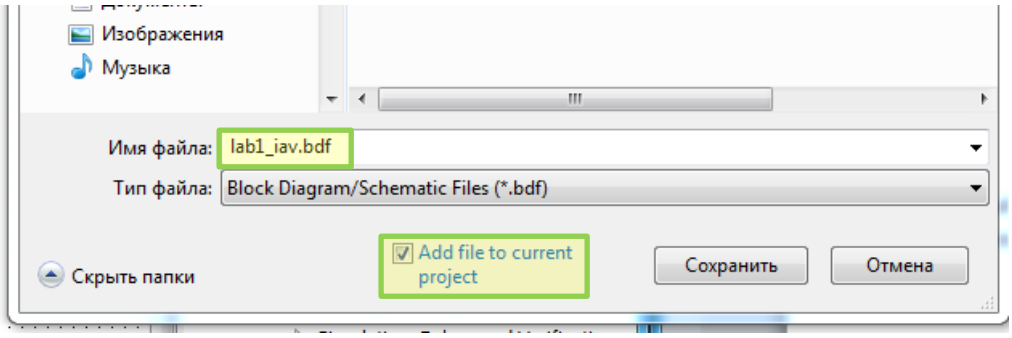


Рис. 25 Сохранение проектного файла

5.3. Нарисуйте схему проекта (рис. 26). Инструменты для создания схемы и работа с ними описаны в документе «Практическое занятие 1_v1». Доступ к созданным экземплярам компонентов из IP catalog-а осуществляется через автоматически создаваемую папку **Project** в библиотеке элементов (рис. 27). Будьте внимательны при указании диапазона номеров в шинных сигналах **Q[27..0]** на выходе счетчика **Counter_28b** и **Q[27..24]** на входе модуля памяти **ROM** и написании сигнала **SW5, SW4** на входе декодера **Decoder_2to4**. Выбирайте верхний регистр при именовании сигнала для лучшей читаемости схемы.

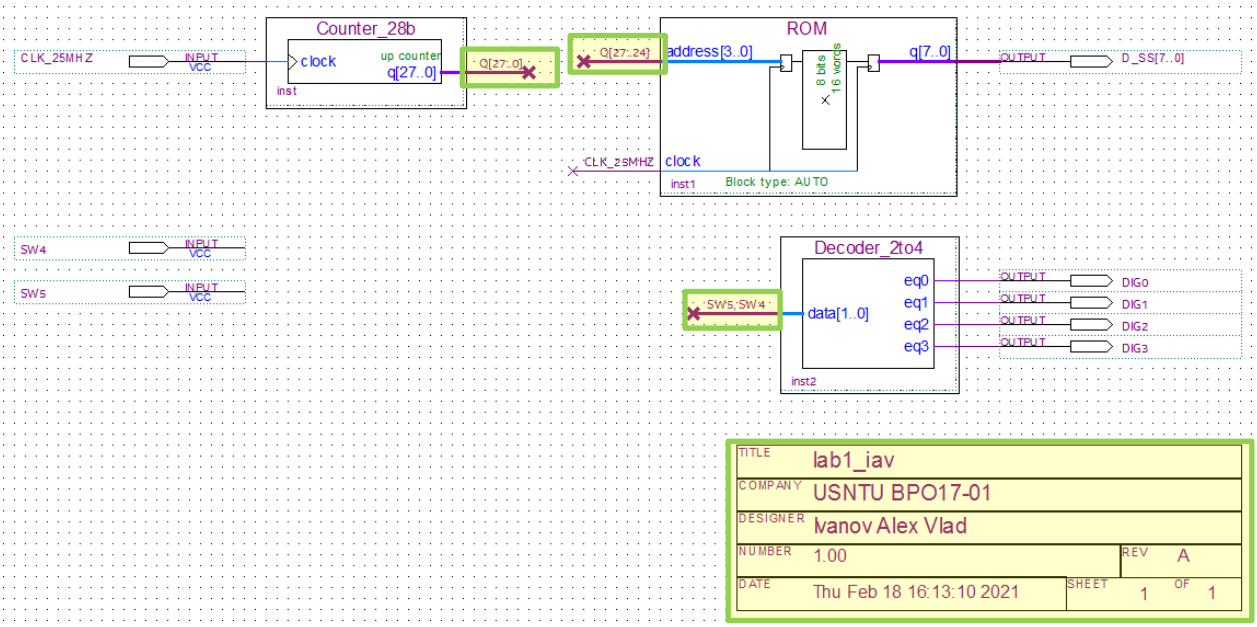


Рис. 26 Схема проекта lab1_iav

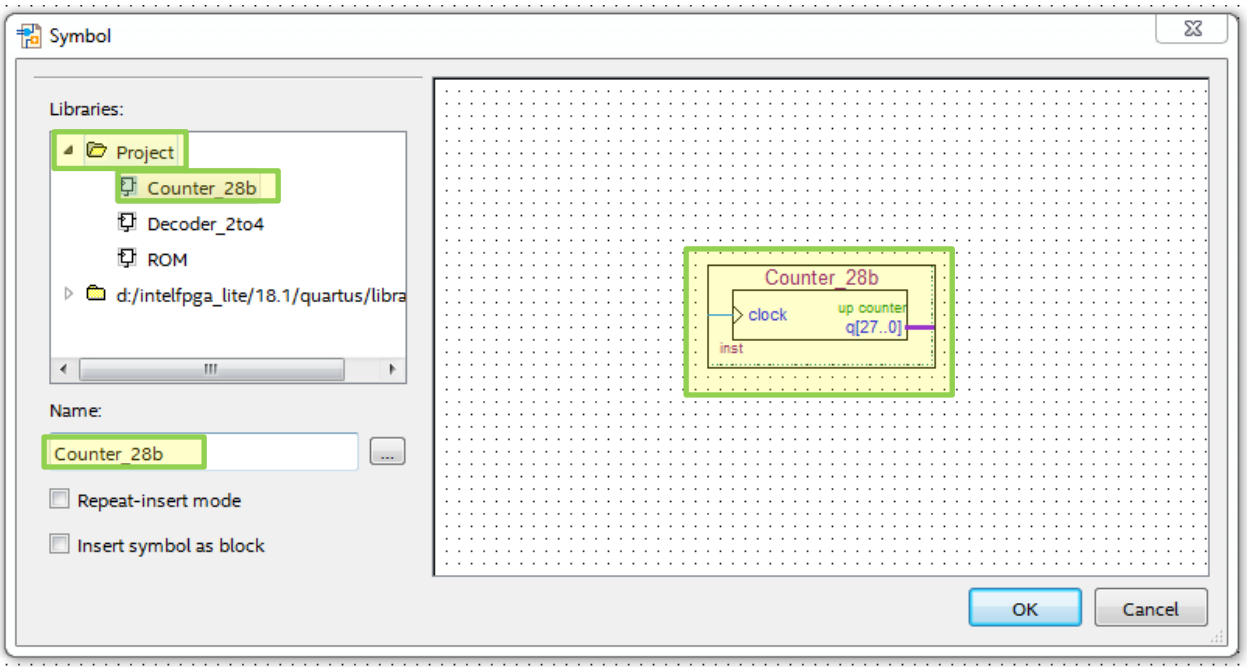


Рис. 27 Папка **Project** в библиотеке элементов

Схему удобней рисовать с включенной сеткой, управление ее видимостью можно выполнить следующим образом: **View-> Show Guidelines** или кликнуть ПКМ на экране редактора**,** затем выбрать **Show->Show Guidelines.** Если на схеме видны только части имен, настройте параметры монитора на вашем компьютере (может быть установлен не тот масштаб изображения). После создания схемы сохраните ее.

6 Назначение выводов ПЛИС

6.1. После того как схема полностью нарисована, можно провести ее проверку на правильность составления и возможность синтезировать в базисе выбранной микросхемы ПЛИС. Для этого выберите задачу **Analysis@Synthesis** из окна задач **Tasks** или нажмите кнопку на панели задач или выполните **Ctrl+K** или **Processing->Start->StartAnalysis@Synthesis.** Запустится компиляция, результаты которой можно наблюдать в окне **Compilation Report** (рис. 28). В случае успешной компиляции в окне **Message** появится информация об ее успешном окончании (рис. 29). В противном случае появятся сообщения об ошибках и предупреждения, которые необходимо устранить. Если схема не установлена в качестве целевого проекта, тогда при компиляции возможна генерация сообщения об ошибке в окне **Messages** следующего типа:

12007 Top-level design entity “lab1_iav” is undefined

Так часто бывает при разработке иерархических проектов, когда приходится отлаживать схемы разного уровня иерархии. Для того чтобы скомпилировать выбранную схему нужно сделать ее проектом верхнего уровня. В этом случае, при открытом окне схемного редактора с нужной схемой, выполните

Project -> Set as top-level entity или **Ctrl+Shift+J**

Текущая схема станет проектом верхнего уровня и может быть скомпилирована.

6.2. В результате компиляции также формируется список сигналов для редактора выводов **Pin Planner**, которые необходимо привязать к выводам микросхемы. Откройте редактор **Pin Planner** (рис. 30) одним из следующих способов: **Assignments -> Pin Planner** / **Ctrl + Shift + N** / кнопка **Pin Planner** на панели задач .

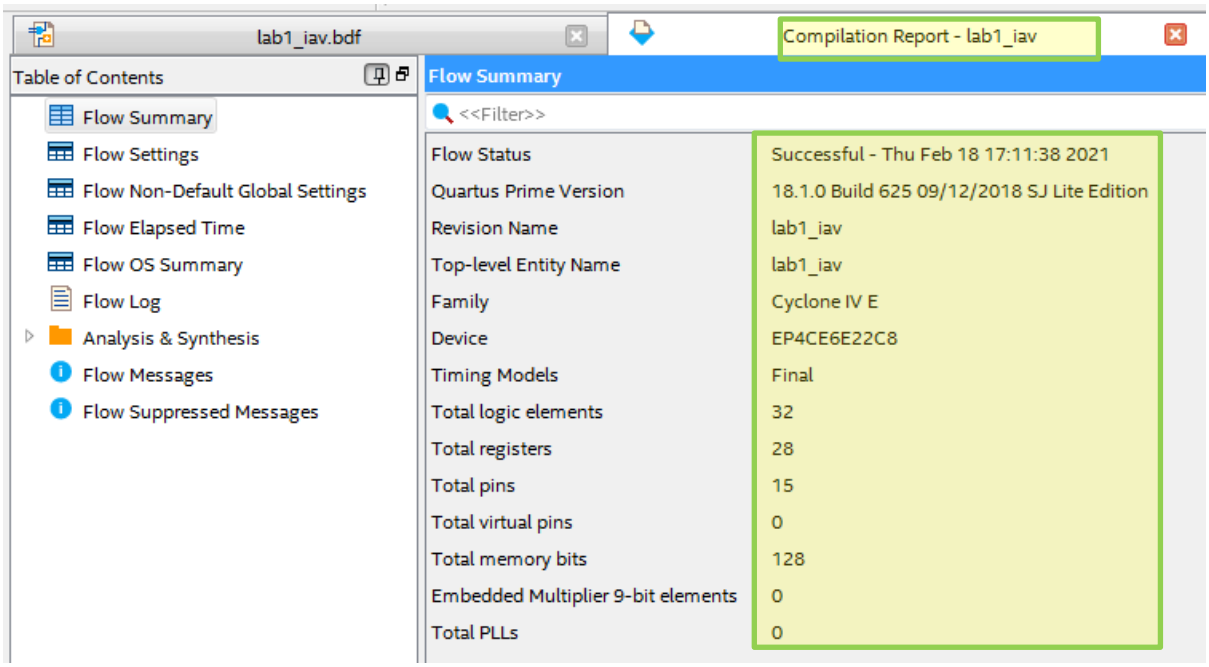


Рис. 28 Окно с результатами компиляции

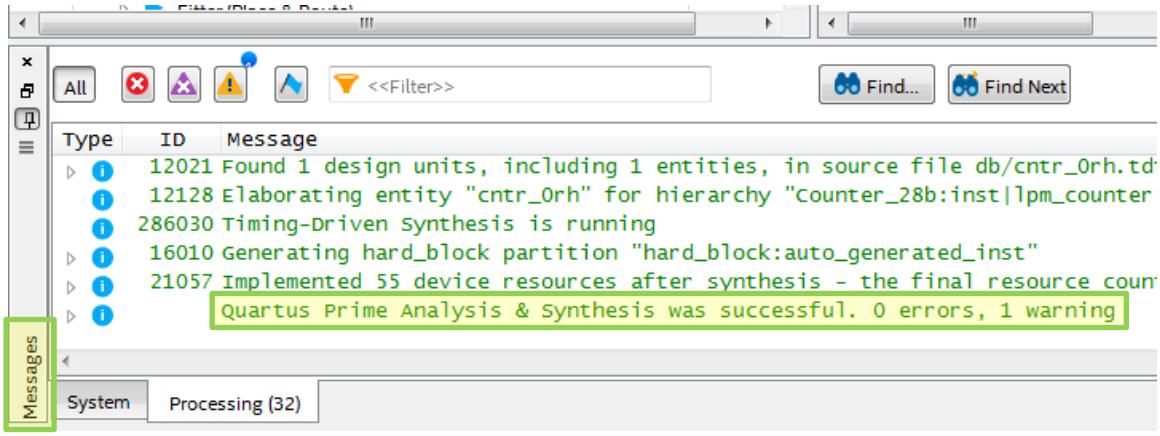


Рис. 29 Окно Message с результатами компиляции

6.3. 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 рисунке 31 (набирайте только номер вывода!). После того как все сигналы будут привязаны к выводам микросхемы ПЛИС, закройте окно редактора, все изменения автоматически сохранятся в файле проекта **Pr1.qsf** (quartus setting file).

6.4. 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** (рис. 32). После назначения выводов у символов ввода-вывода на схеме появятся номера выводов (**PIN_nn**) микросхемы, к которым они привязаны.

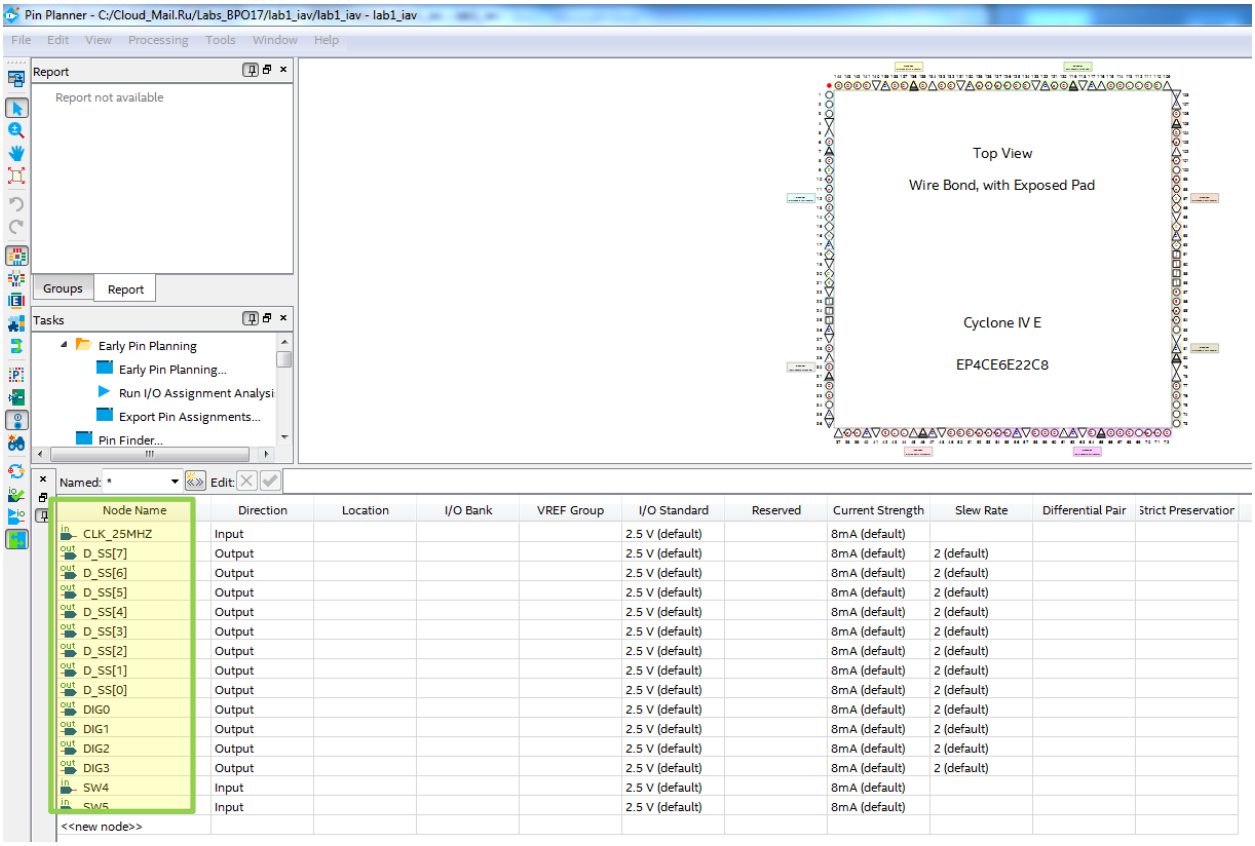


Рис.30 Окно редактора Pin Planner

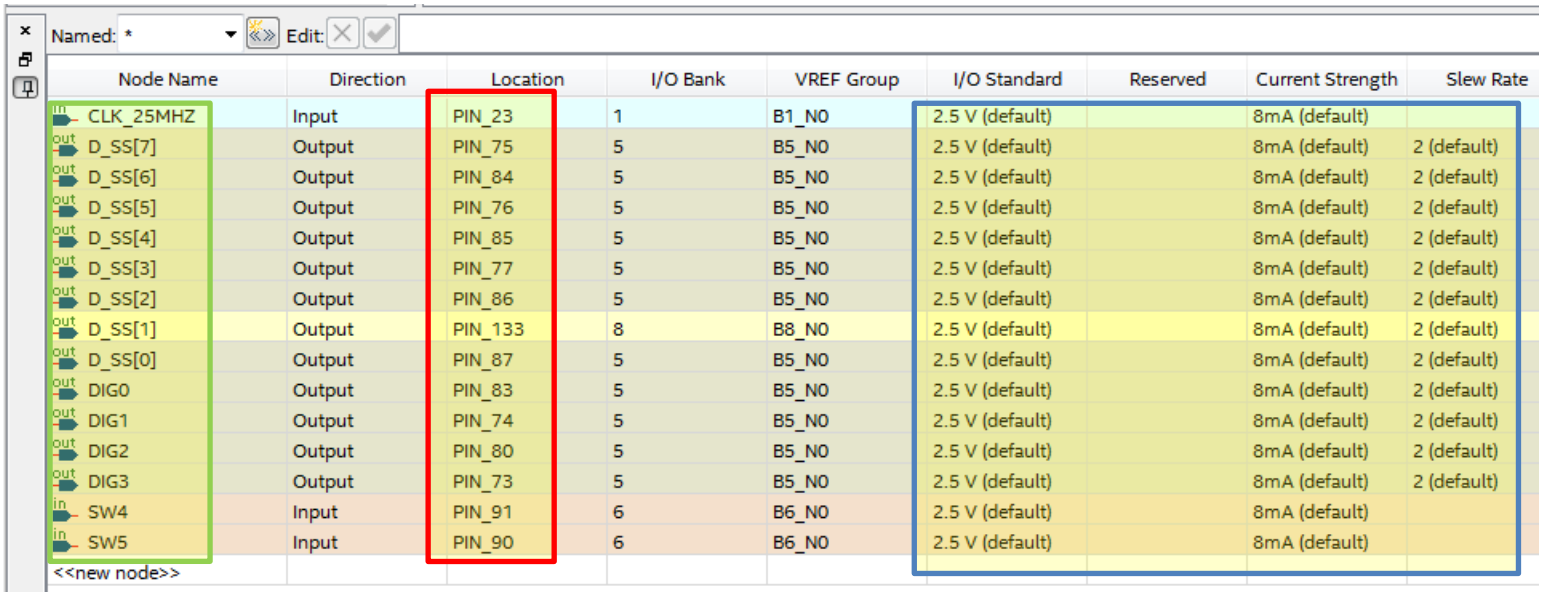


Рис.31 Окно редактора **Pin Planner** с привязкой сигналов проекта к выводам ПЛИС

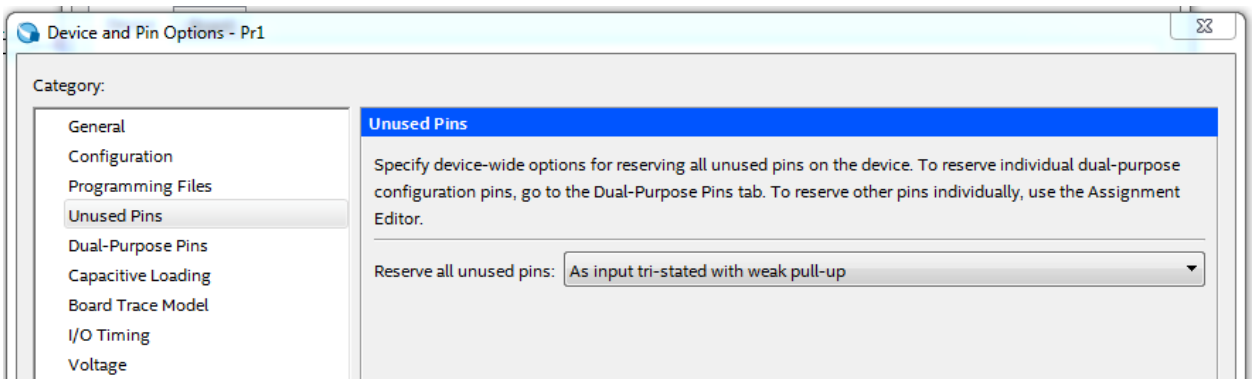


Рис.32 Назначение неиспользуемых выводов ПЛИС в режим **“As input tri-stated with weak pull-up”**

7. Создание sdc – файла

7.1. Создайте файл с информацией о временных требованиях (т.е. на какой частоте проект должен работать гарантированно) к проекту (рис. 33):

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

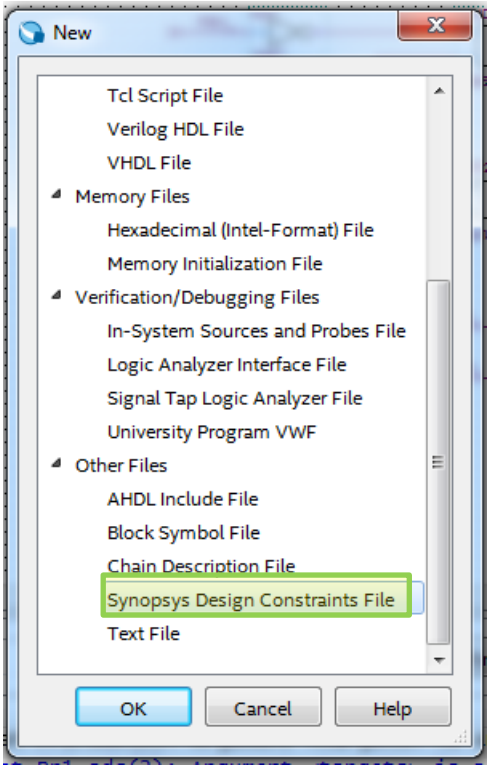


Рис. 33 Создание файла с временными требованиями

7.2. Сохраните созданный текстовый проектный файл **File =>Save As** с именем ***lab1_iav.sdc*** **.** Минимальный набор команд, который должен быть в нем определен, показан на рисунке 34.

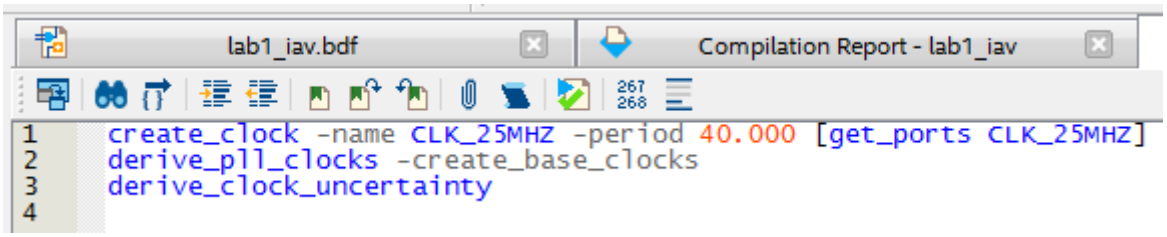


Рис. 34 Задание временных требований к проекту

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 на отладочной плате.

7.3. Сохраните файл и проследите, чтобы файл появился в окне **Project Navigator** (Выберите вкладку **Files**)(рис. 35).

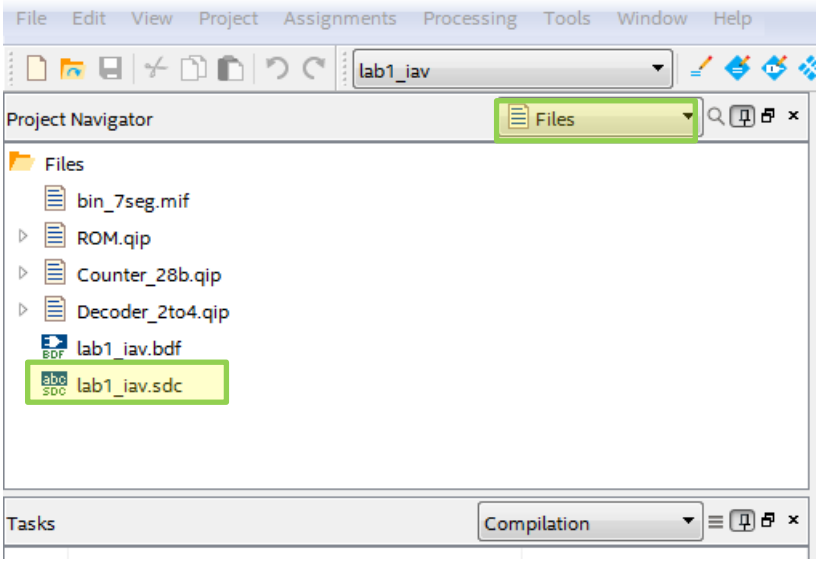


Рис. 35 Перечень файлов проекта в окне **Project Navigator**

В противном случае, при открытом окне с файлом ***lab1_iav.sdc*** выполните **Project->Add Current File to Project.** Файл ***lab1_iav.sdc*** добавится в проект.

8. Выполнение полной компиляции проекта

8.1. 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. 36). В основном 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 и умножителей).

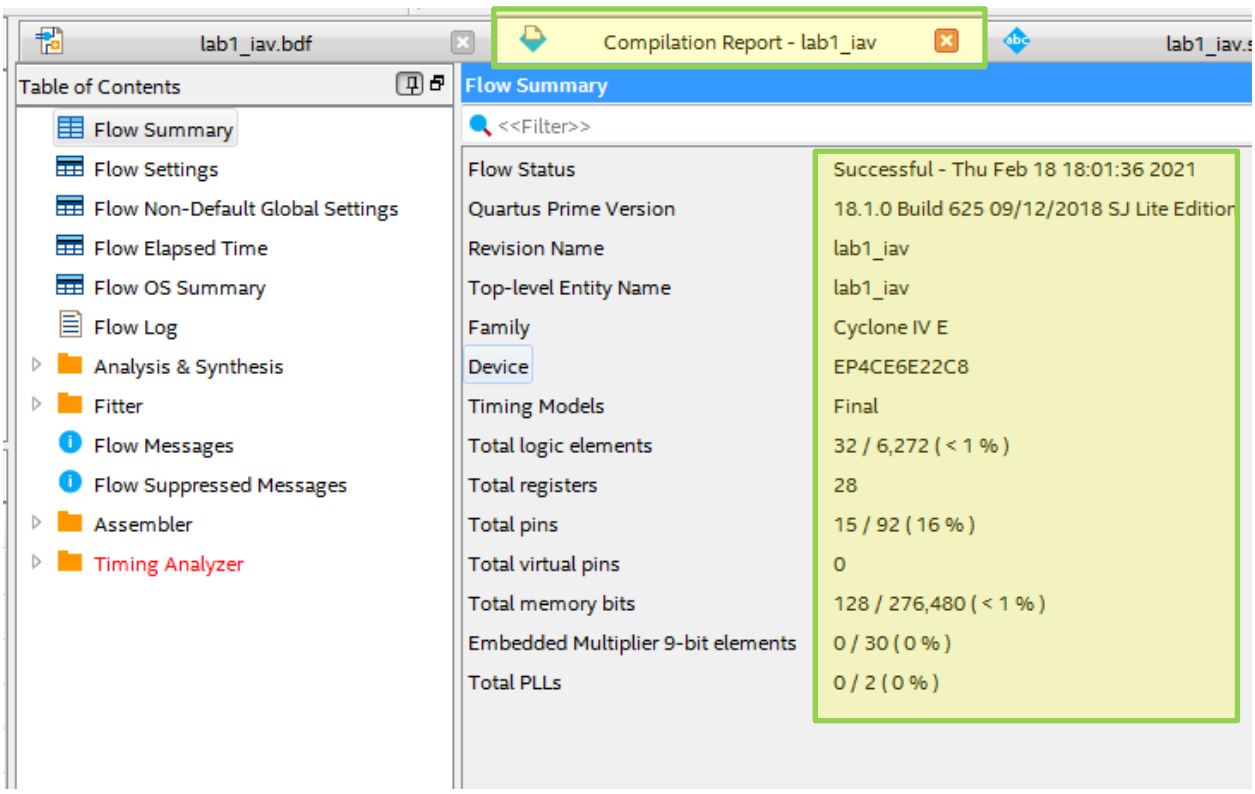


Рис. 36 Окно **Compilation Report** с результатами полной компиляции

8.2. В случае успешной компиляции в окне **Message** появится информация об ее успешном окончании (рис. 37). В противном случае появятся сообщения об ошибках и предупреждения, которые необходимо устранить.

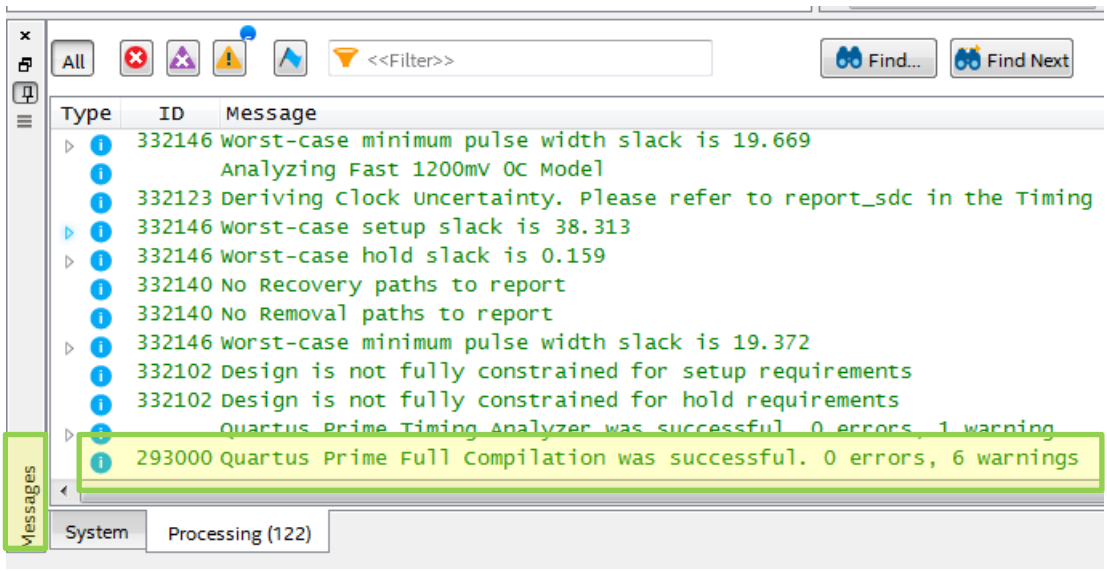


Рис. 37 Окно **Message** с сообщением о выполнении успешной полной компиляции

8.3. В окне **Compilation Report** откройте файл **Fmax Summary** с результатами временного анализа (рис. 38).

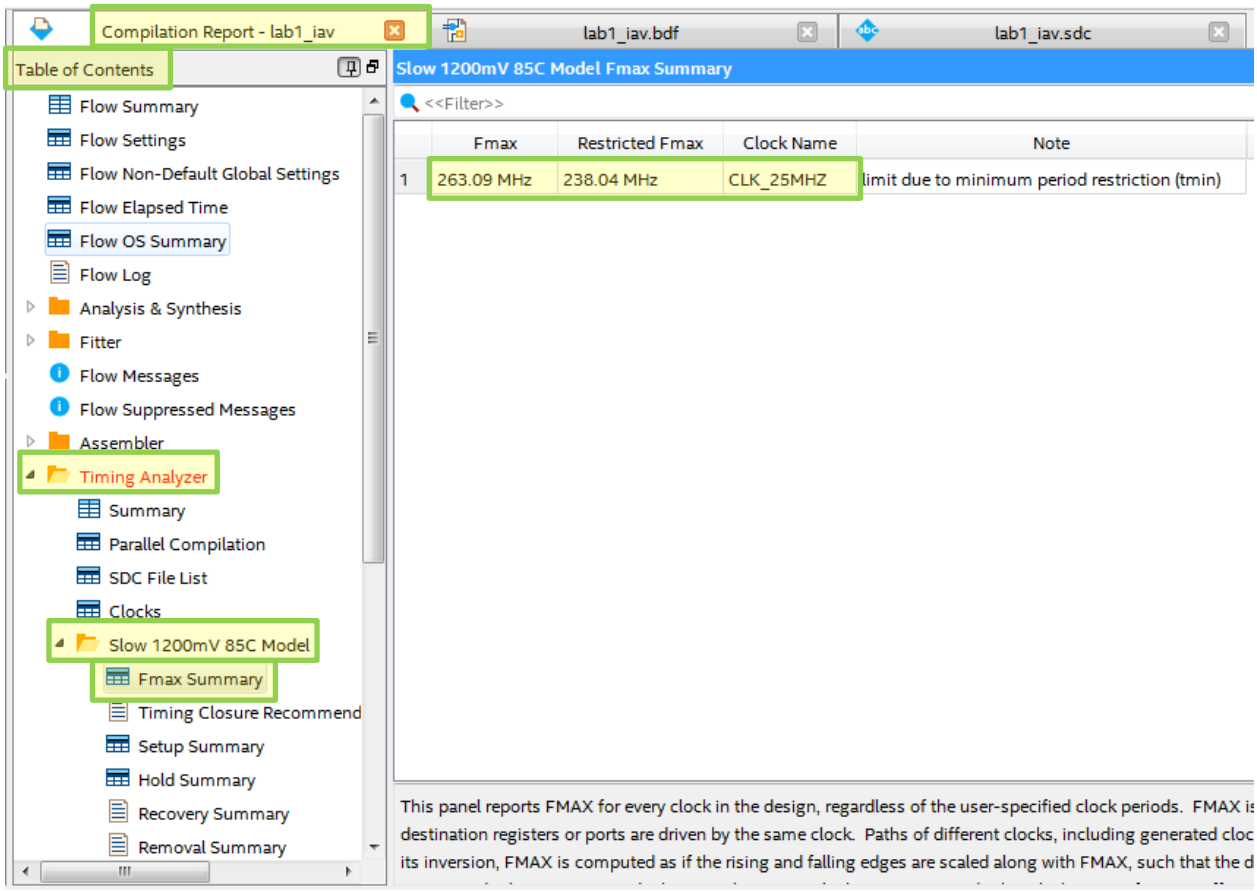


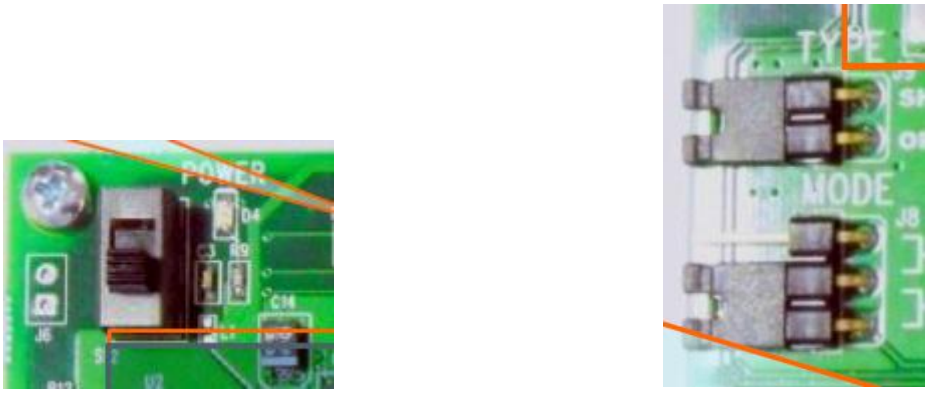
Рис. 36 Окно **Compilation Report** с результатами временного анализа проекта

Убедитесь, что максимально возможная частота работы проекта (Fmax), превышает заданные в файле **lab1_iav.sdc** временные требования (CLK_25MHZ).

9. Конфигурирование ПЛИС и проверка работоспособности на отладочной плате.

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

9.2. Подключите отладочную плату **miniDLab–CIV** к компьютеру с помощью **USB** кабеля (тип A–miniB), и включите питание (рис. 37a). Проверьте правильность установки перемычек (рис. 37b).



a b

Рис. 37 Переключатель Power(a) и конфигурация разъемов установки режимов работы платы (b)

9.3. Откройте окно программатора **Programmer** из окна **Tasks (Program Device)** или кнопкой или **Tools->Programmer** (рис. 38).

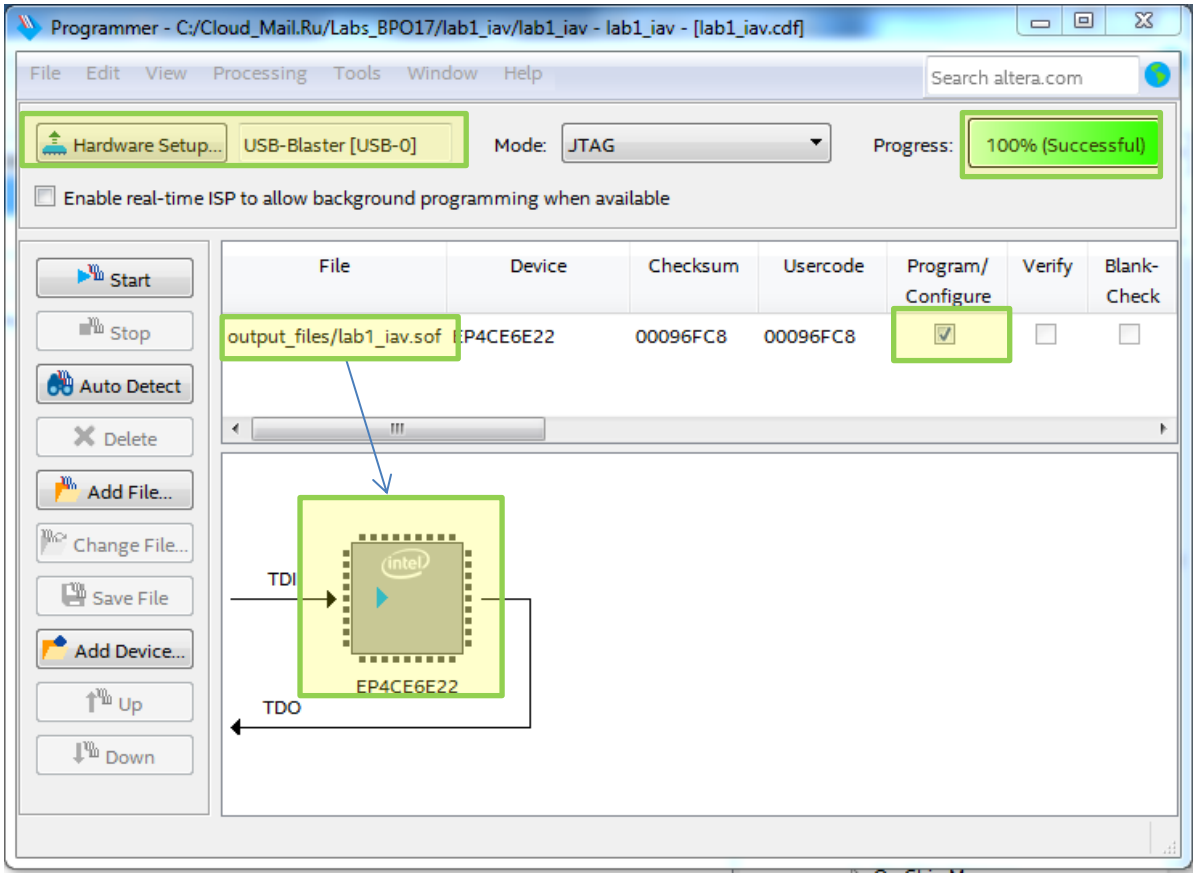


Рис.38 Окно программатора

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

9.5. 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)** ***lab1_iav.sof*.**

9.6. Выполните **Start**, в окошке **Progress** будет виден текущий статус загрузки.

9.7. После успешной загрузки конфигурации ПЛИС проверьте работу проекта (табл. 1).

Таблица 1

Состояние переключателей SW4, SW5	Поведение схемы
00	На 1 разряд 7-сегментного индикатора выводятся последовательно символы 0, 1, 2, 3, …, a, b, c, d, e, f, 0, ...
01	На 2 разряд 7-сегментного индикатора выводятся последовательно символы 0, 1, 2, 3, …, a, b, c, d, e, f, 0, ...
10	На 3 разряд 7-сегментного индикатора выводятся последовательно символы 0, 1, 2, 3, …, a, b, c, d, e, f, 0, ...
11	На 4 разряд 7-сегментного индикатора выводятся последовательно символы 0, 1, 2, 3, …, a, b, c, d, e, f, 0, ...

9.8. Выключите и отсоедините отладочную плату от компьютера.

Лабораторная работа завершена.

Общие требования к отчету:

1. Отразите 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.