**ОСНОВНЫЕ ЭТАПЫ ПРОЕКТИРОВАНИЯ В САПР QUARTUS II**

Quartus II позволяет легко реализовать требуемую логическую схему в ПЛИС.

Проектирование в САПР Quartus II включают следующие этапы:

• **Ввод проекта (Design Entry)** –желаемая схема проекта задается либо графическим способом, либо с использованием языков описания аппаратных средств, таких как Verilog HDL ,VHDL и др.

• **Синтез (Synthesis)** – вводимый проект синтезируется в схему, которая состоит из логических элементов (ЛЭ) и логических блоков (ЛБ) в микросхеме ПЛИС.

• **Функциональное моделирование (Functional Simulation)** – синтезируемая схема тестируется на предмет корректности функционирования во встроенном симуляторе, который моделирует зависимость состояния (выбранных разработчиком) сигналов схемы от времени. Это моделирование не учитывает временные задержки сигналов (называемые *логическими гонками*) между логическими элементами микросхемы ПЛИС.

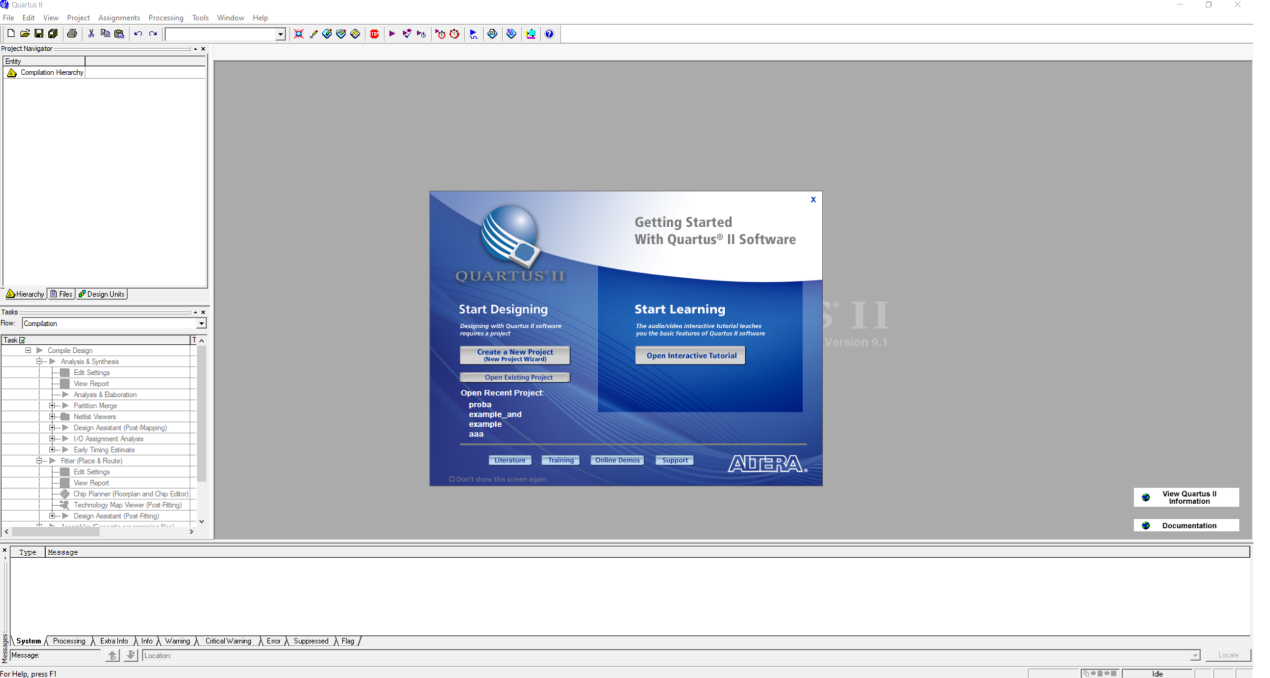
• **Временное моделирование (Timing Simulation)** – схема тестируется для проверок функциональности и временных ограничений, но в отличие от функционального моделирования, здесь для обнаружения наличия или отсутствия логических гонок учитываются реальные задержки выбранных сигналов.

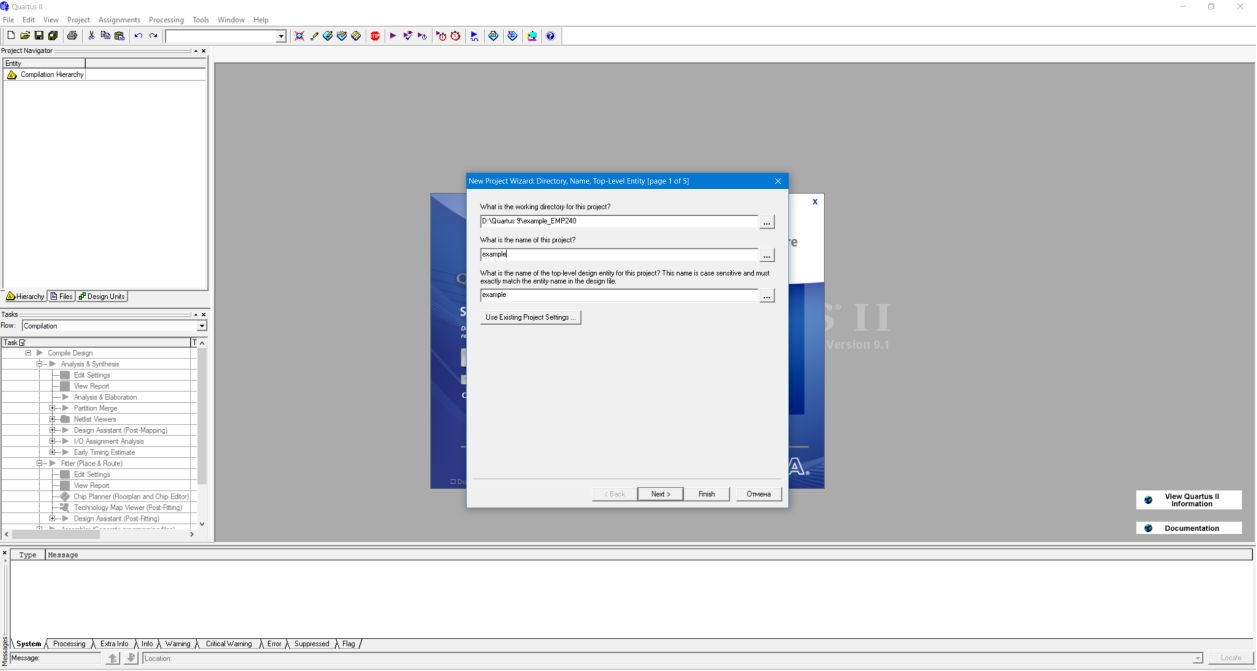
• **Программирование и конфигурация (Programming and Configuration)** - разработанная схема размещается в микросхеме ПЛИС путем программирования электронных связей между конфигурируемыми логическими элементами, что реализуется путем передачи конфигурационного файла с компьютера либо в микросхему ПЛИС, либо в дополнительную (не встроенную в ПЛИС) память, если такая имеется в отладочном комплекте.

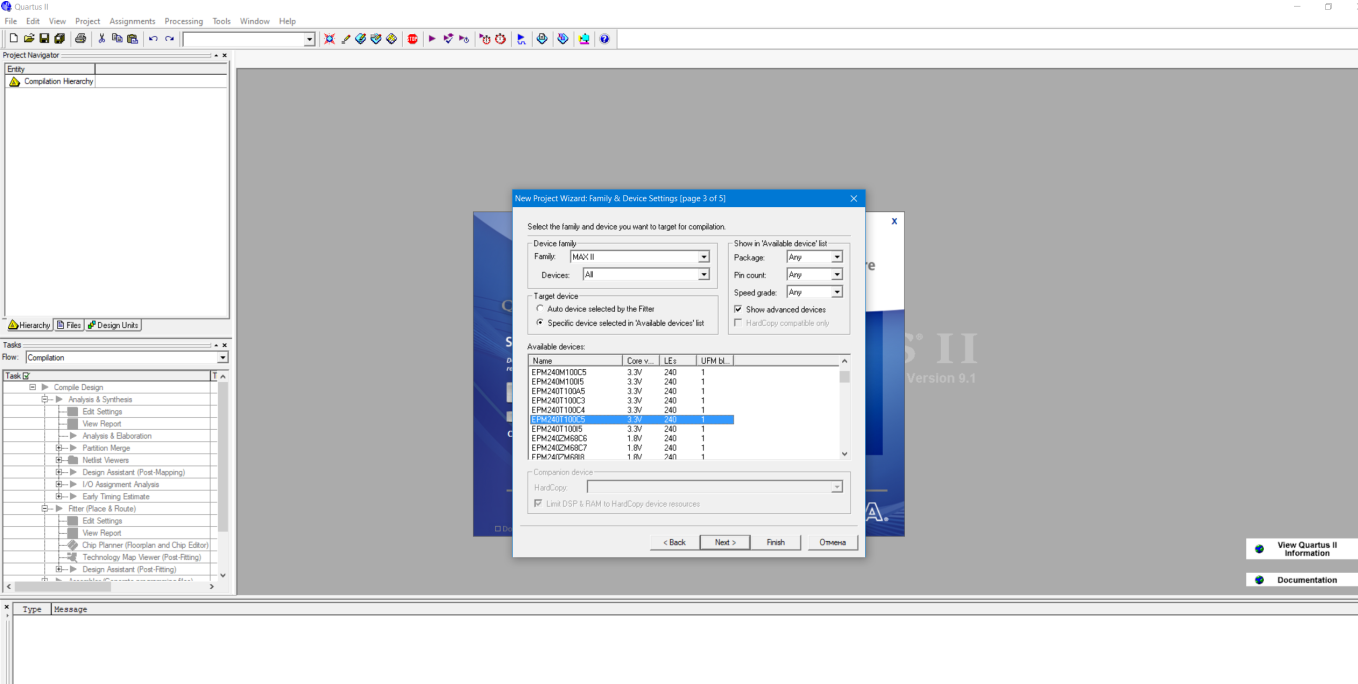
Применение отмеченных выше этапов будет рассмотрено отдельно на примере проектирования простой логической схемы. Для этого необходимо выполнить следующую последовательность действий:

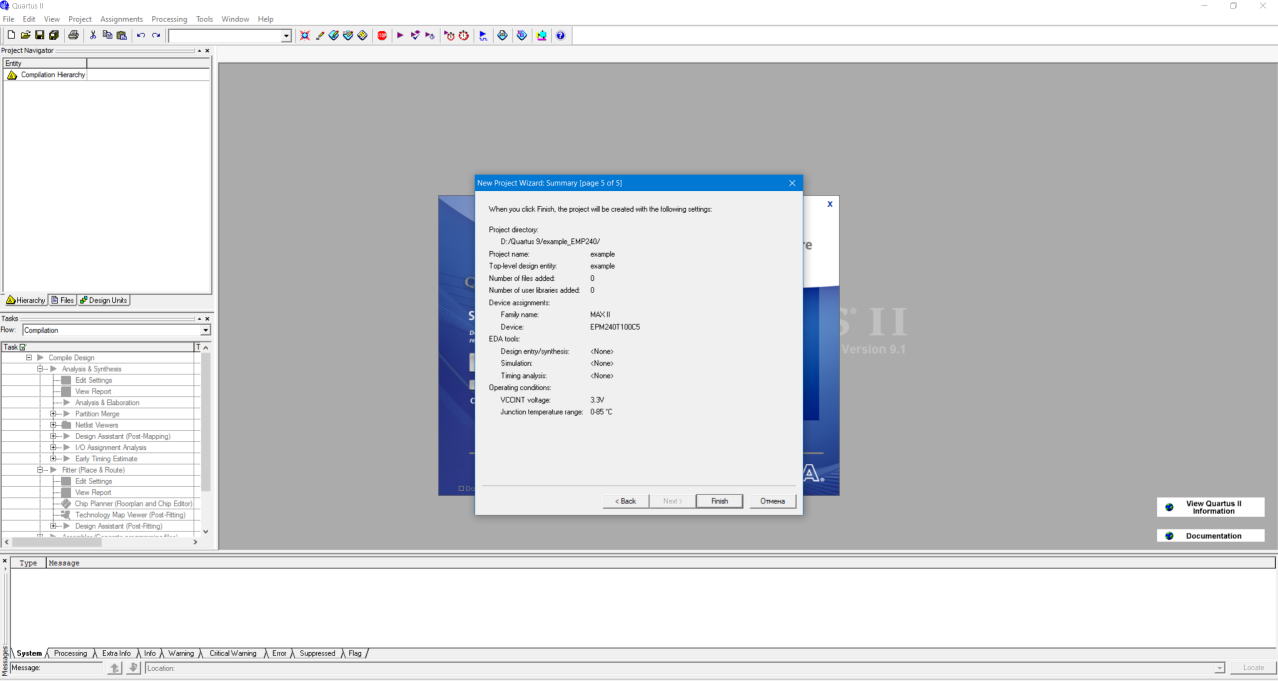
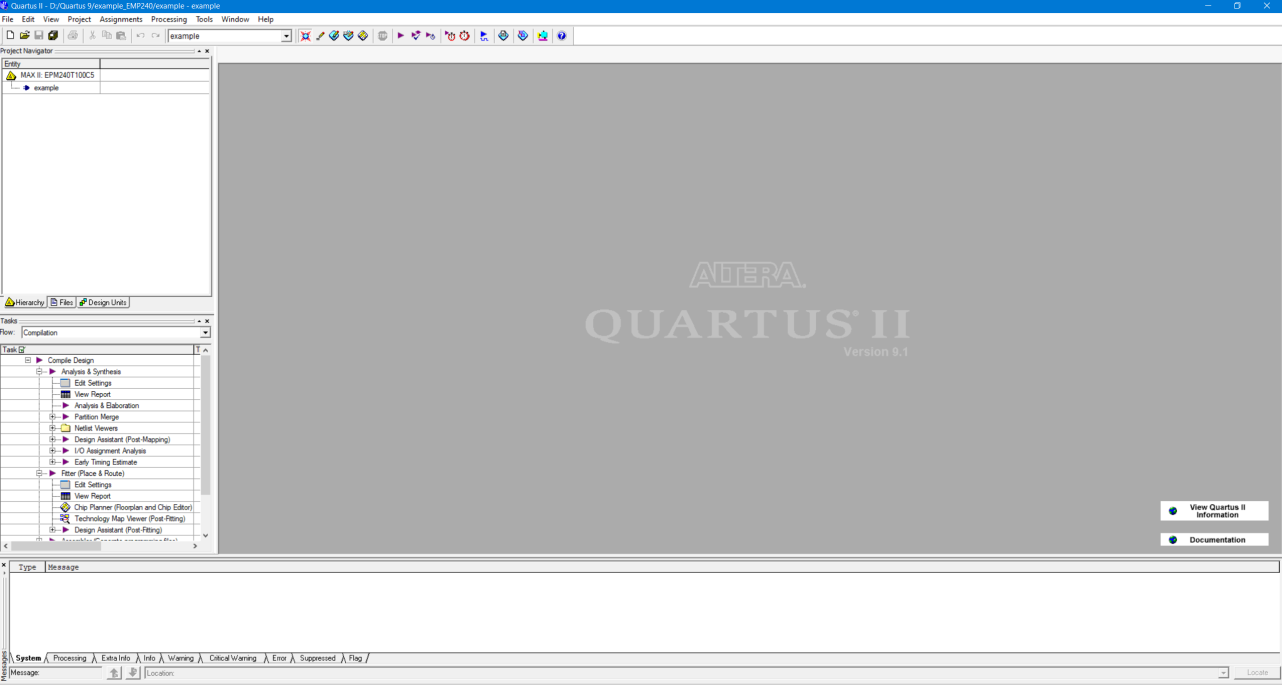
* создание проекта;
* ввод логической схемы;
* синтез цепей в схеме;
* моделирование разработанной логической схемы;
* назначение входов и выходов схемы на выбранной микросхеме ПЛИС;
* программирование и настройка микросхемы ПЛИС на плате.

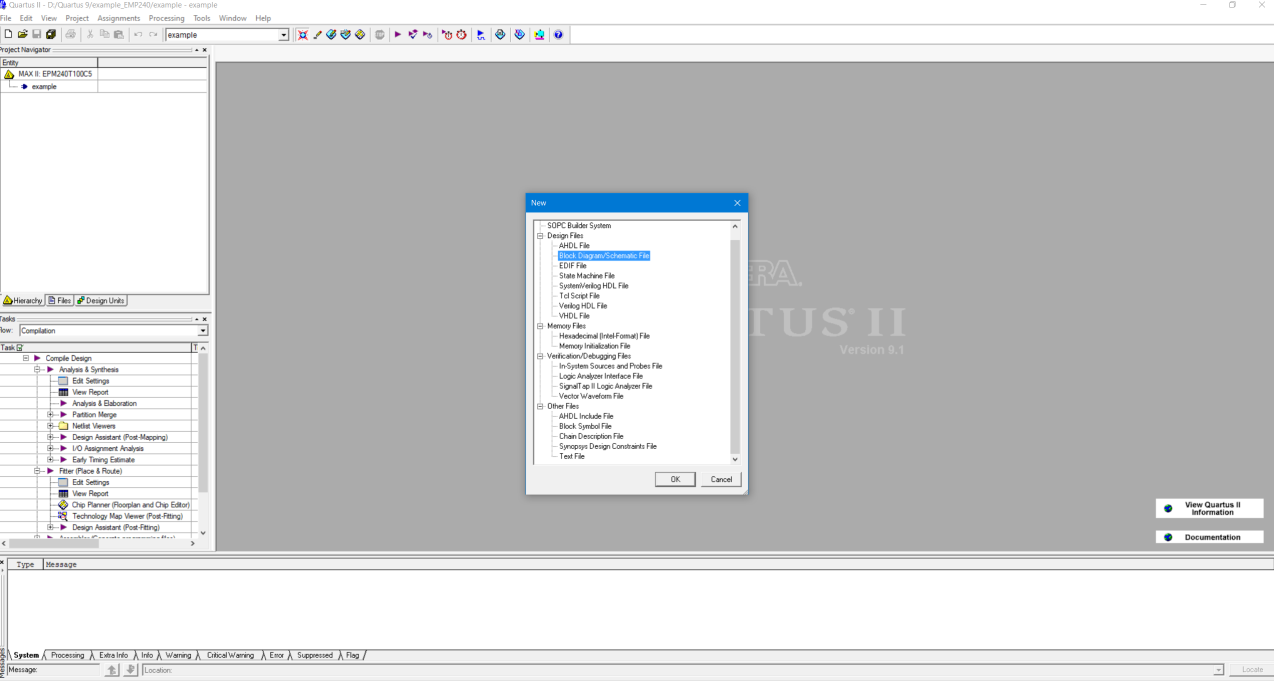
**Создание (Design Entry) и синтез (Synthesis) проекта для логической схемы**

В начальном окне работы с Quartus II выбираем один из двух режимов: «Создание нового проекта» или «Открыть существующий проект».

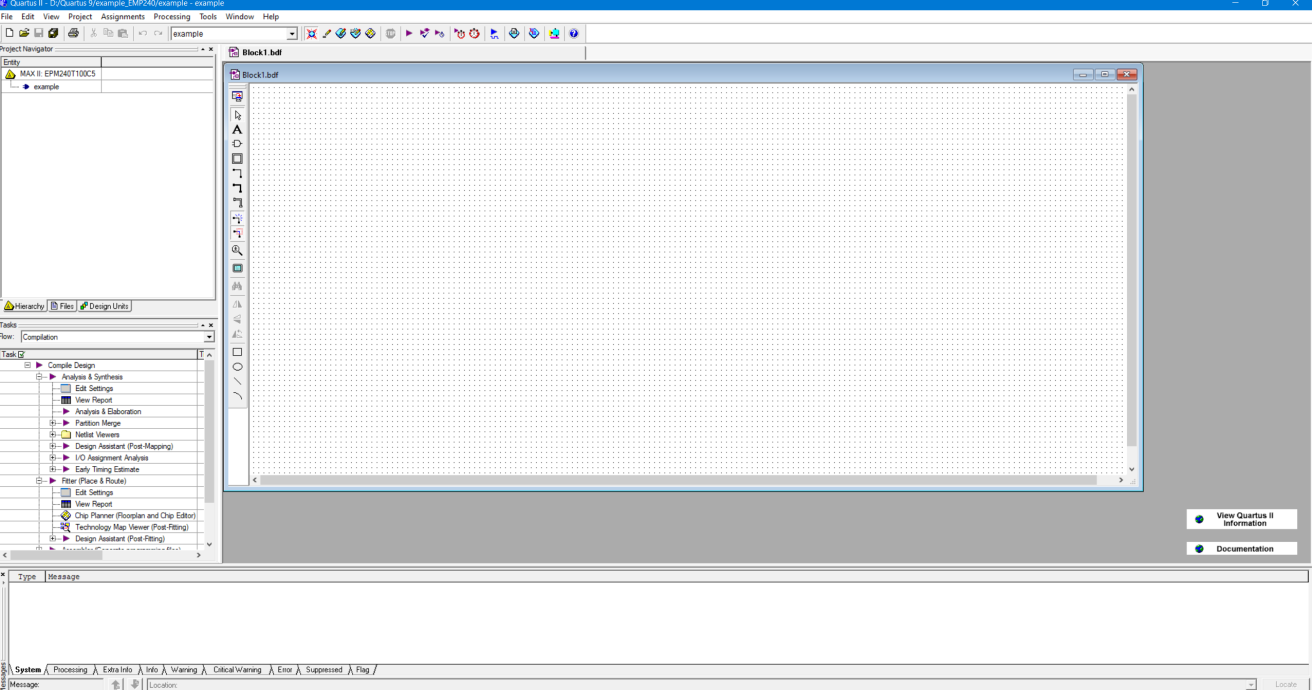
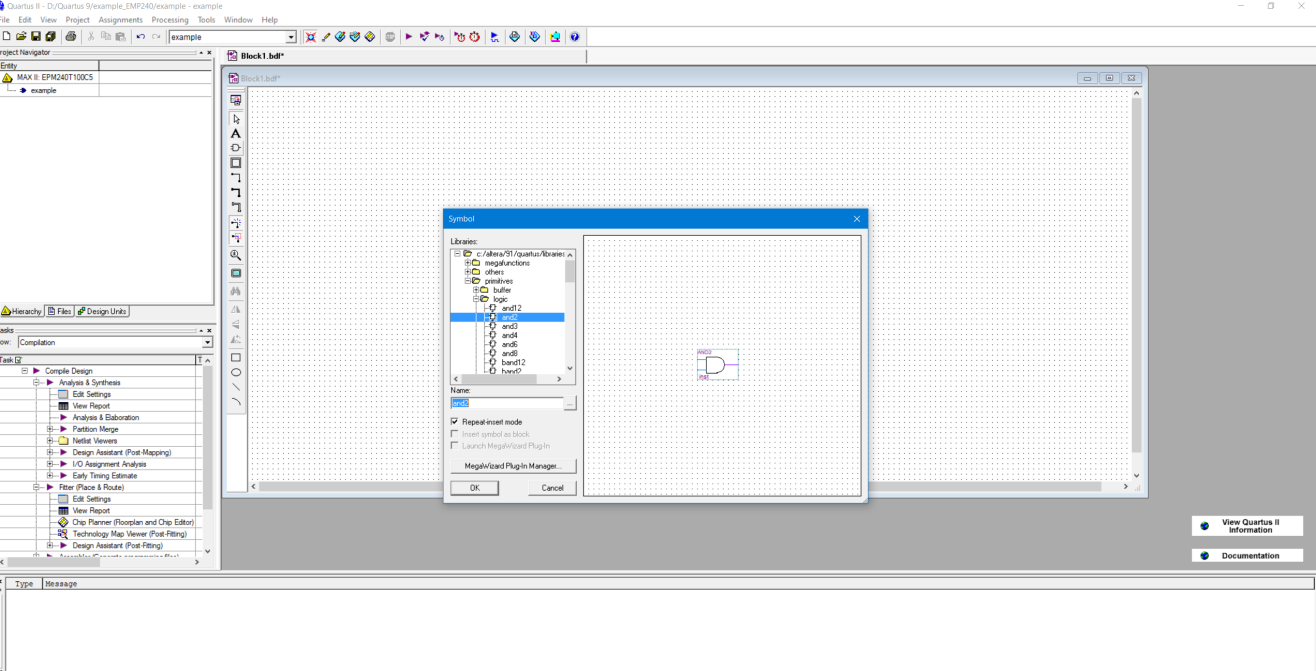
Для создания нового проекта выбираем папку и указываем название проекта.

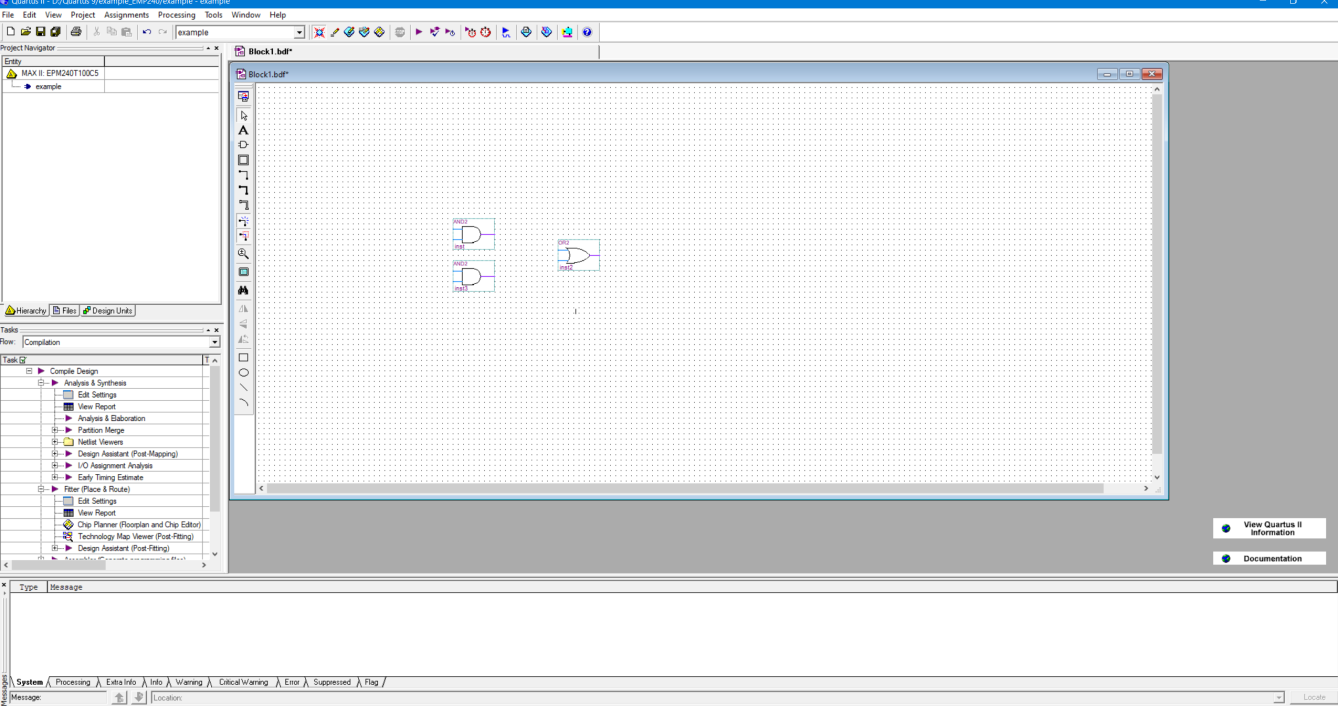
 Далее нажимаем Next, в следующем окне вновь Next. В новом окне выбираем плату, для которой будет создаваться проект. В нашем случае это будет плата MAX II с чипом EPM240T100C5. Затем нажимаем Next. В следующем окне опять Next и далее Finish.

 В результате выполненных действий будет создан пустой проект:

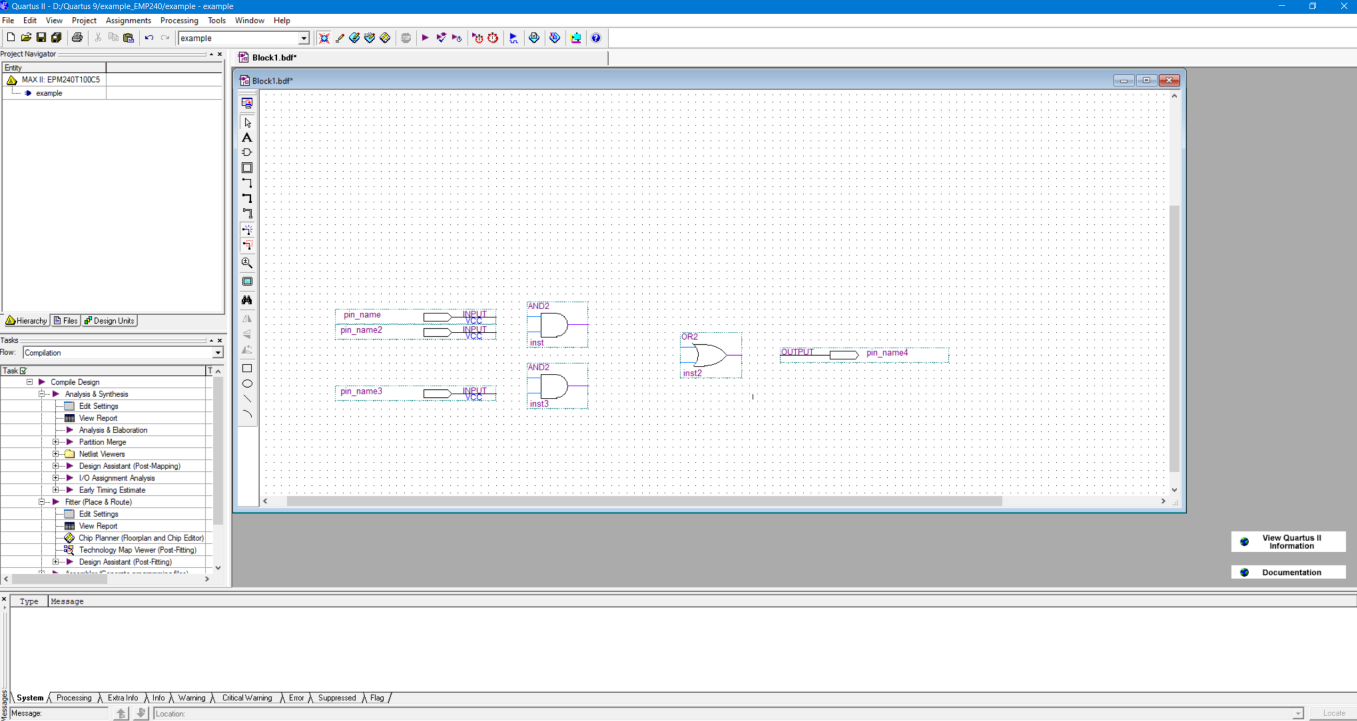
Далее приступаем к созданию схемы: File -> New. В выпавшем меню выбираем пункт Block Diagram/Schematic File.

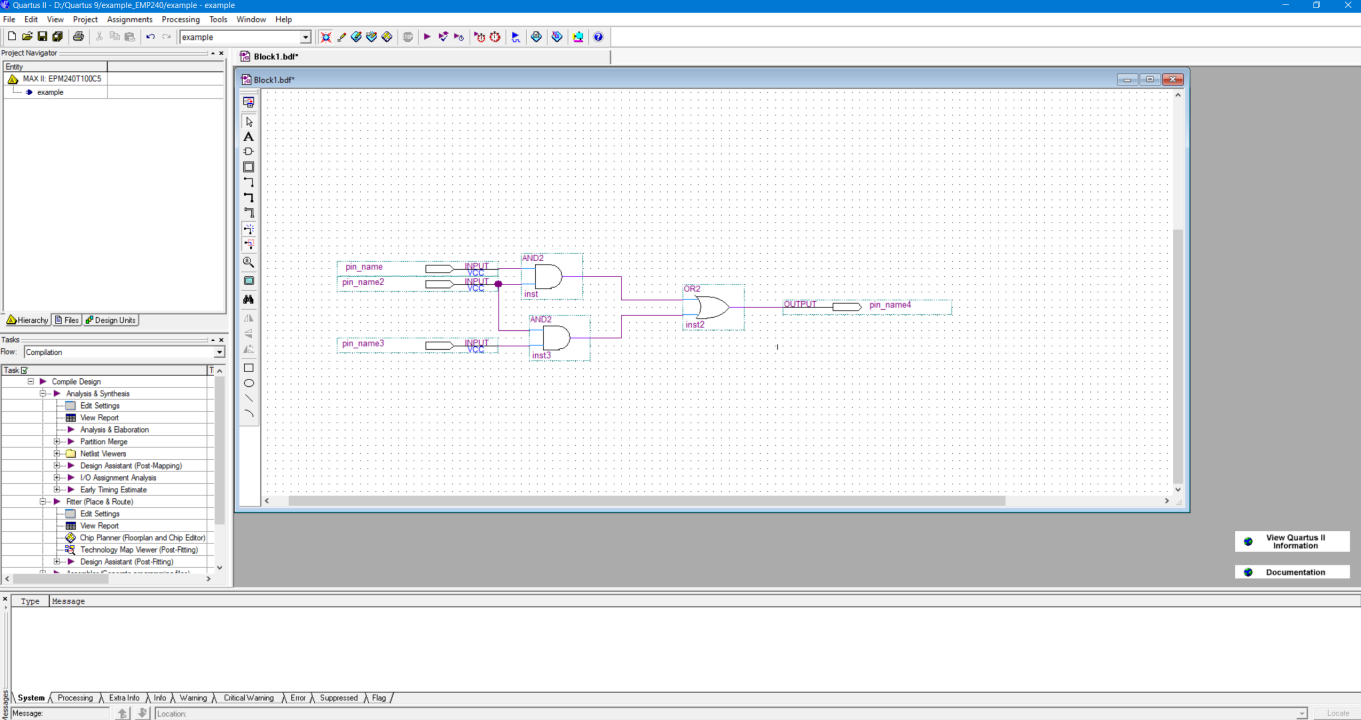
В центре расположено поле для создания схемы справа от него расположена палитра с инструментами для создания схемы. Выбираем в ней элемент Symbol Tool.

Далее в библиотеке логических элементов primitives - logic выбираем требуемый элемент схемы и нажимаем Ok.

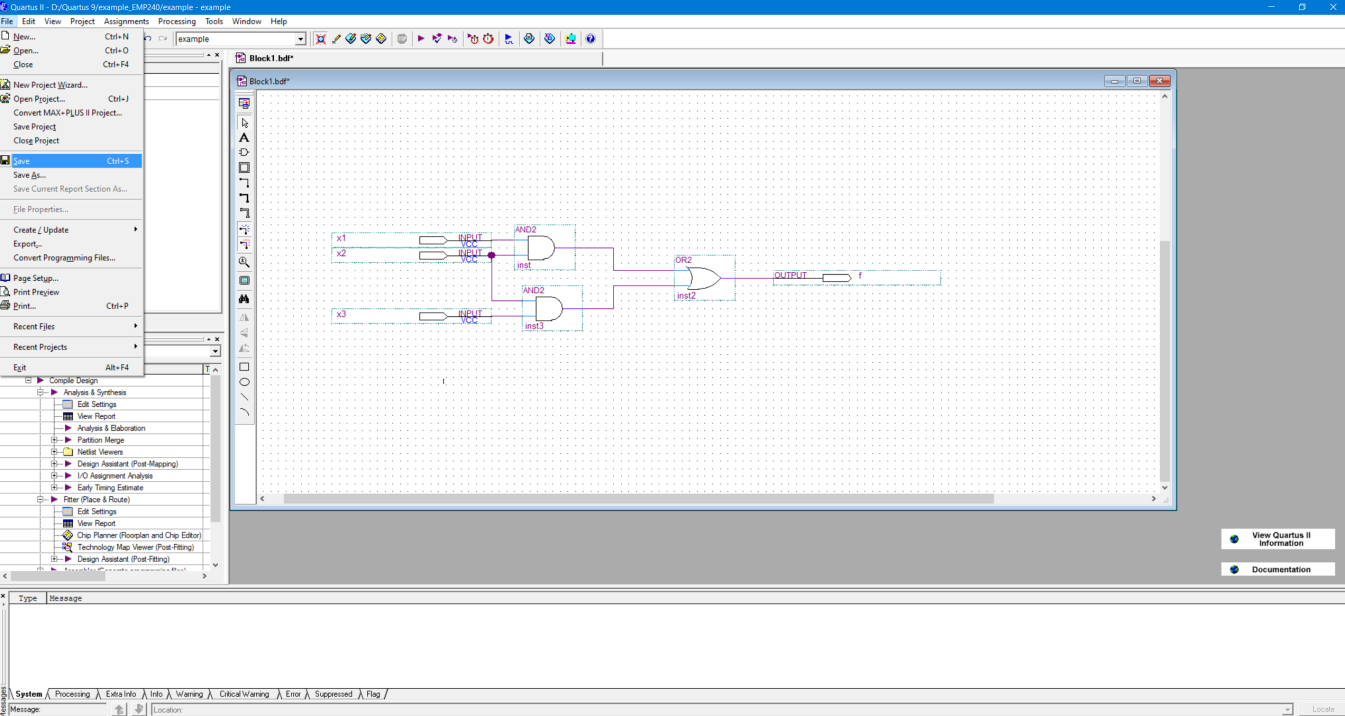
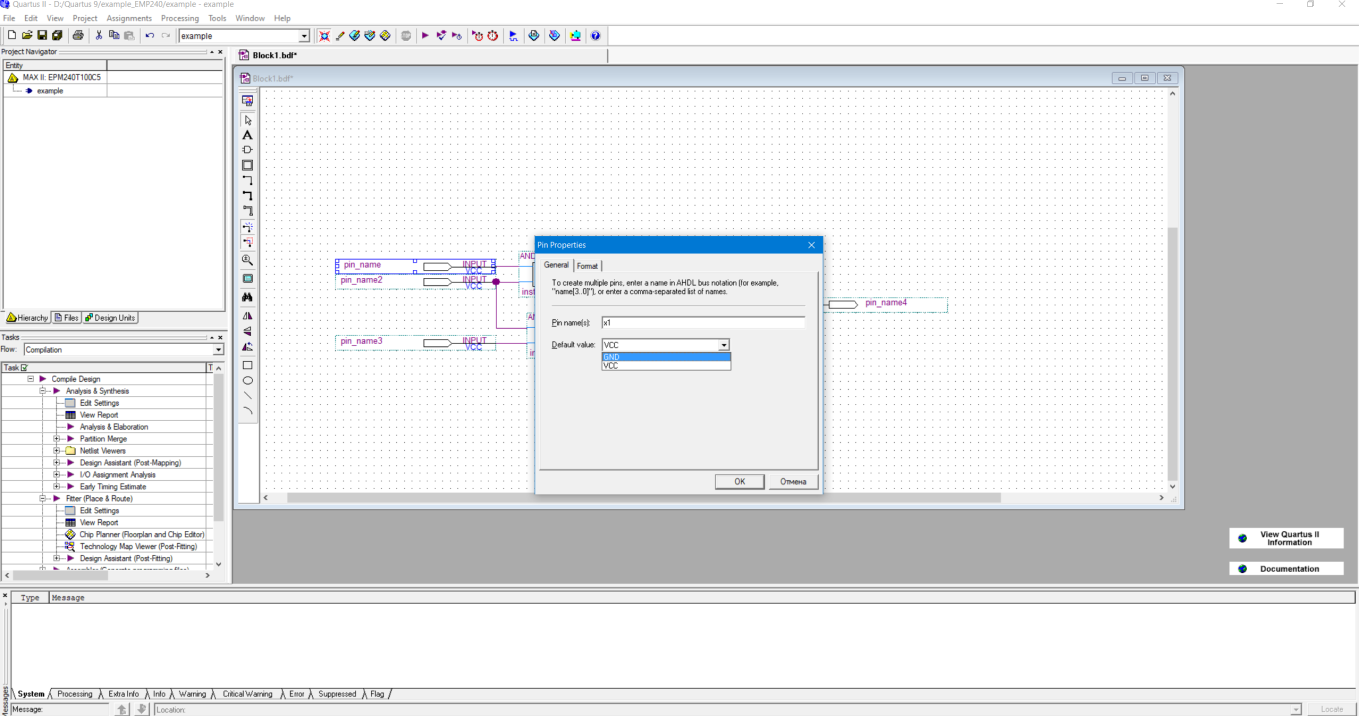
Последовательно выбирая из списка требуемые элементы и добавляя их на рабочий стол схемы формируем список элементов схемы. Например, схема, реализующая функцию:

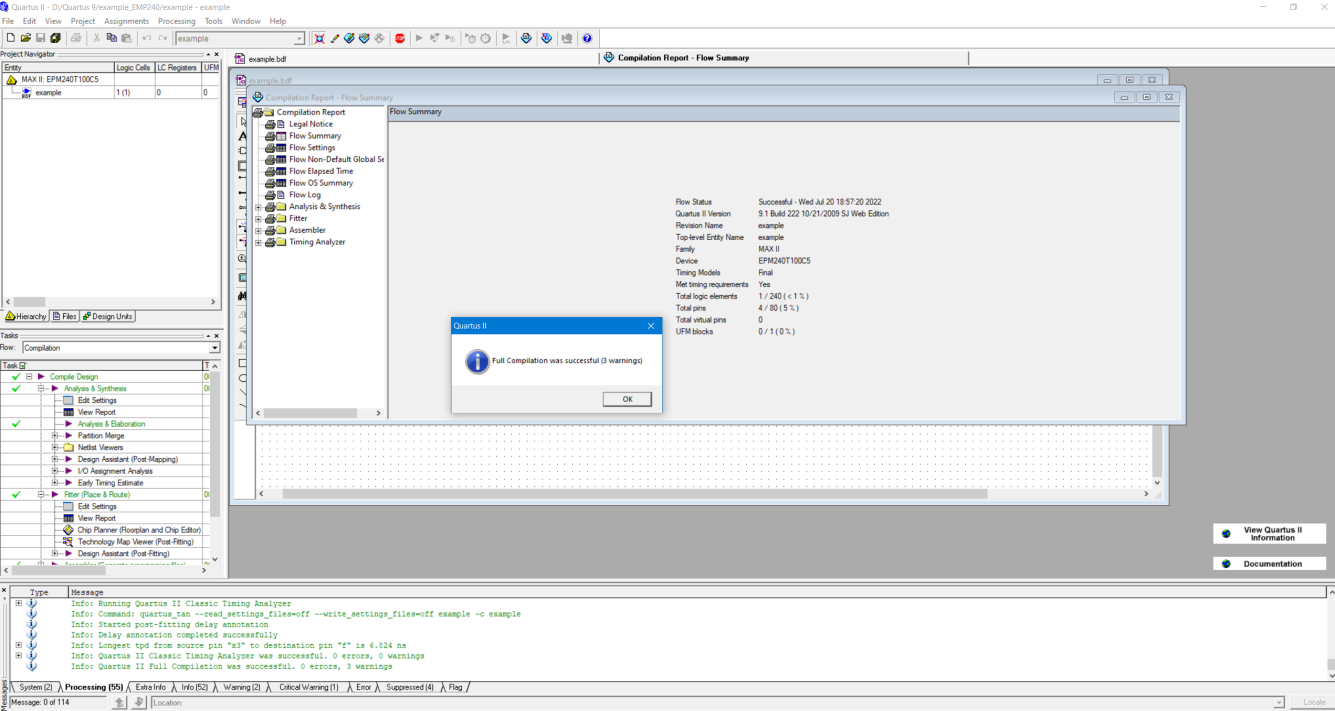
f(x1,x2,x3)=x1x2 + x2x3 , будет содержать три элемента.

 Далее, разместив все элементы схемы, переходим к добавлению в схему входов и выходов. Для этого выбираем primitives - pin-input или primitives - pin-output соответственно для входных и выходных контактов.

Выбрав таким образом элементы требуемые для реализации функции, переходим к указанию всех связей на схеме согласно реализуемой логической функции. Для этого наводим курсор на pin начала связи, нажимаем на левую кнопку мыши и перемещаем указатель на pin конца связи.

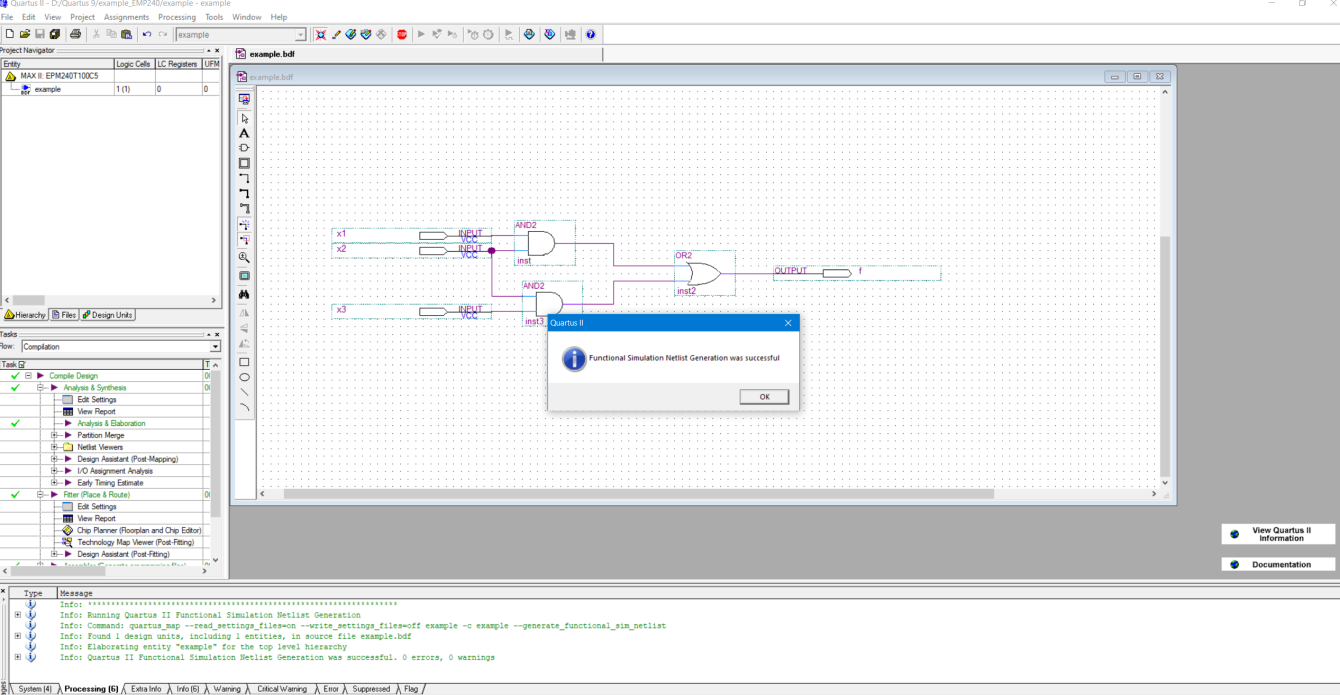
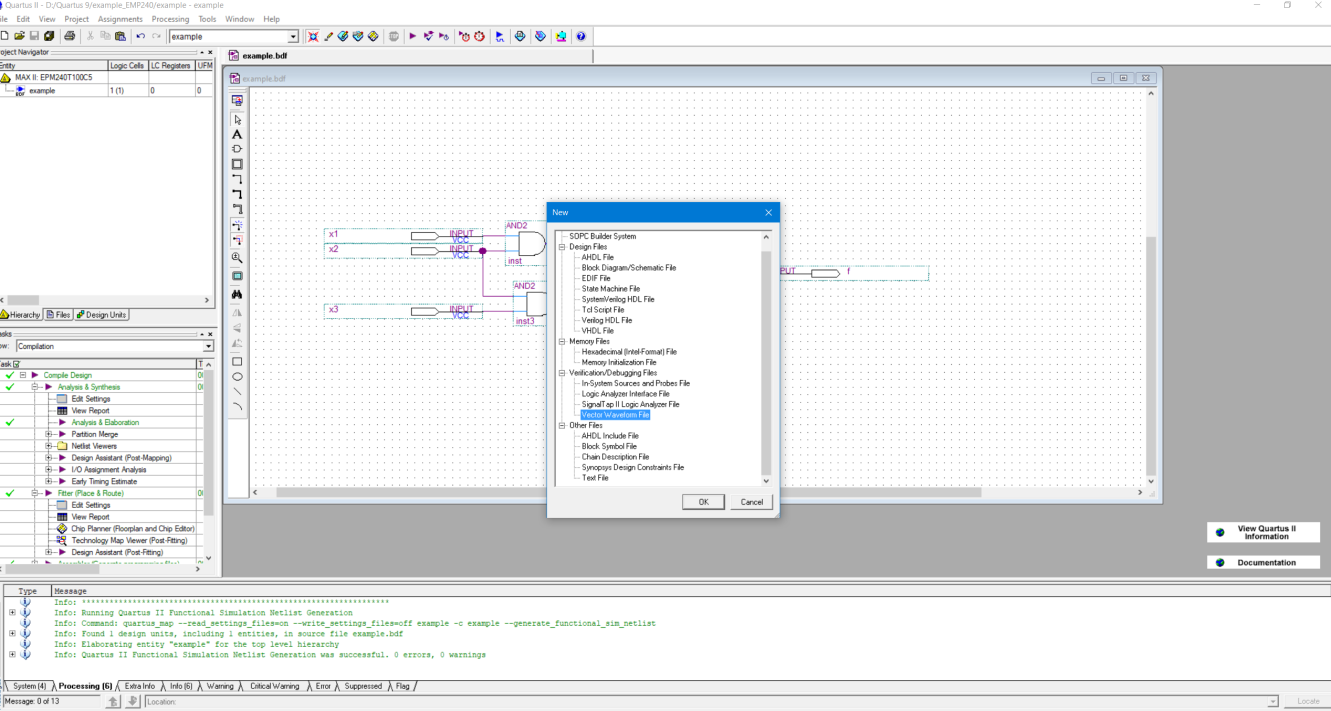
Можно заменить названия входных и выходных pin\_name на названия входных переменных (x1, x2 и x3) и функции f. Для этого выбираем pin\_name и двойным щелчком входим в окно, в котором можно задать новое имя для контакта и значение на нем по умолчанию VCC или GND.

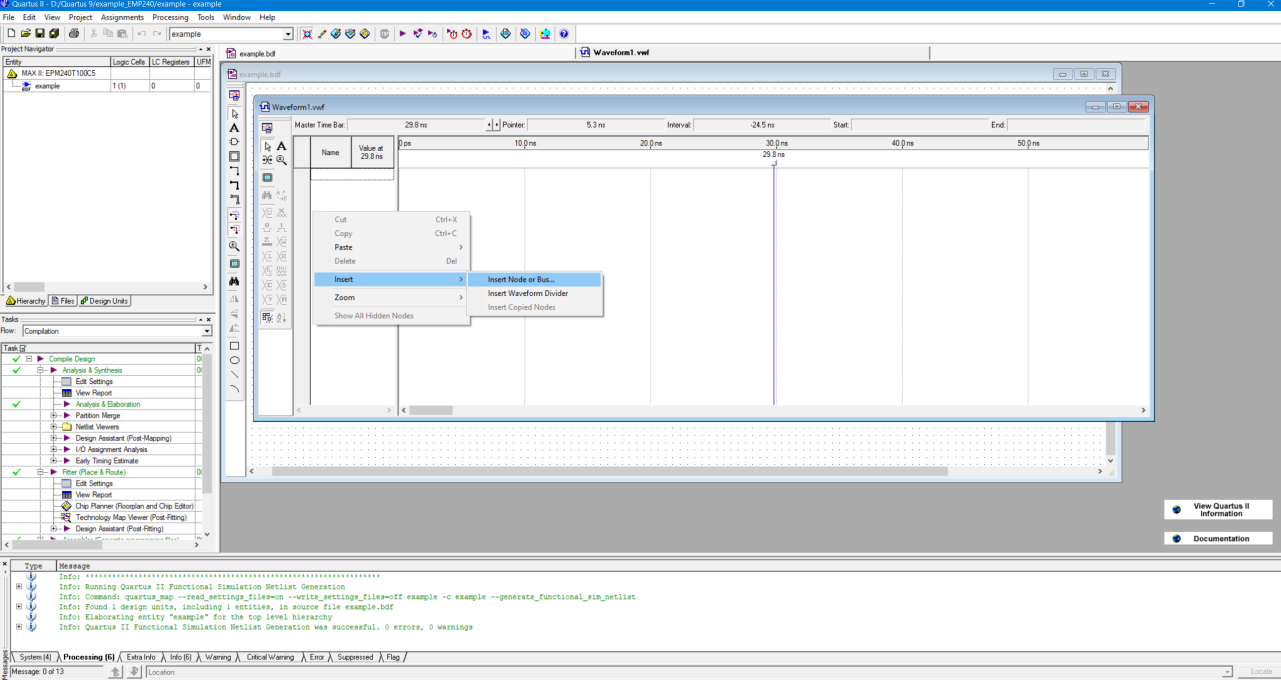
Функциональная схема для функции f построена. Далее сохраняем проект: File->Save и для построенной схемы выполняем компиляцию проекта: Processing->Start Compilation.

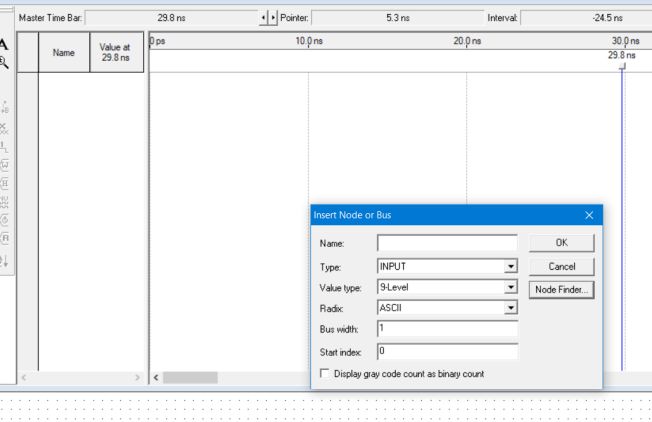
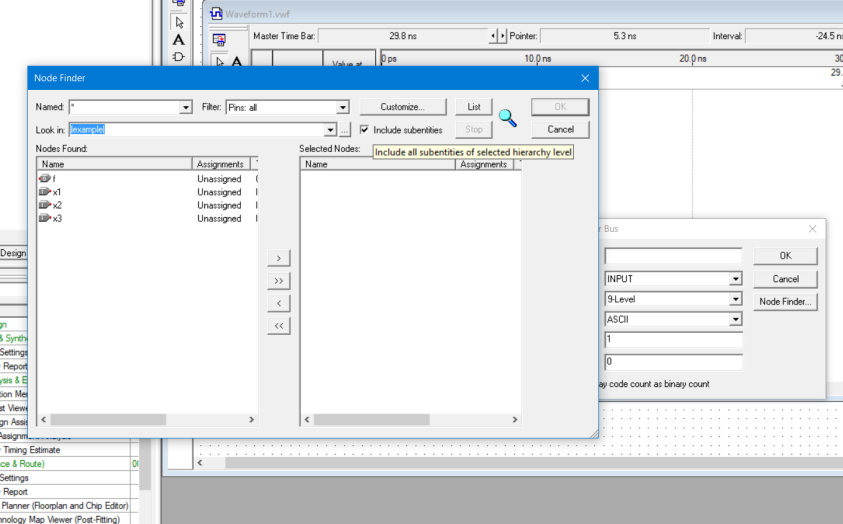
Итак, компиляция проекта выполнена успешно.

**Функциональное моделирование (Functional Simulation) логической схемы**

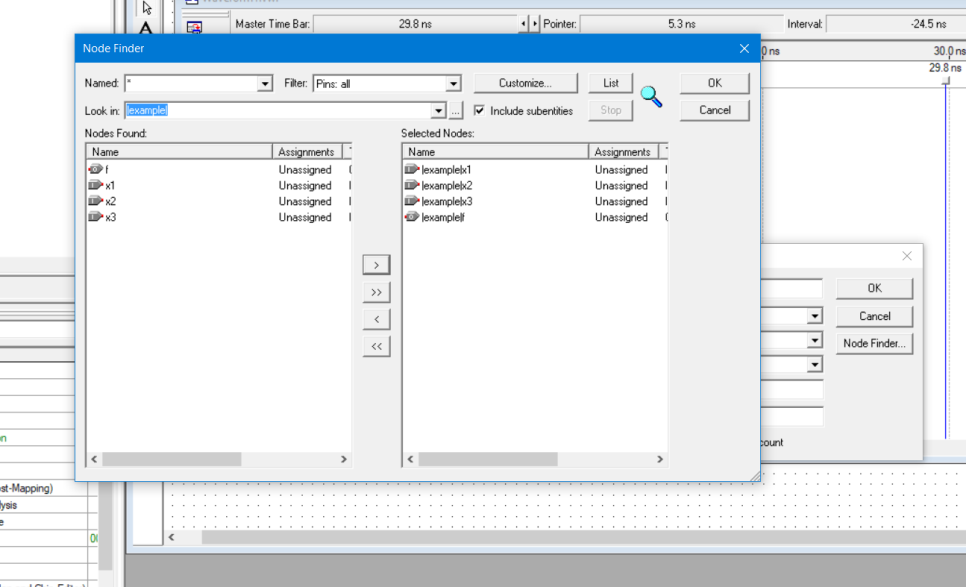
Далее выполняем пункт меню: Processing->Generate Functional Simulational Netlist.

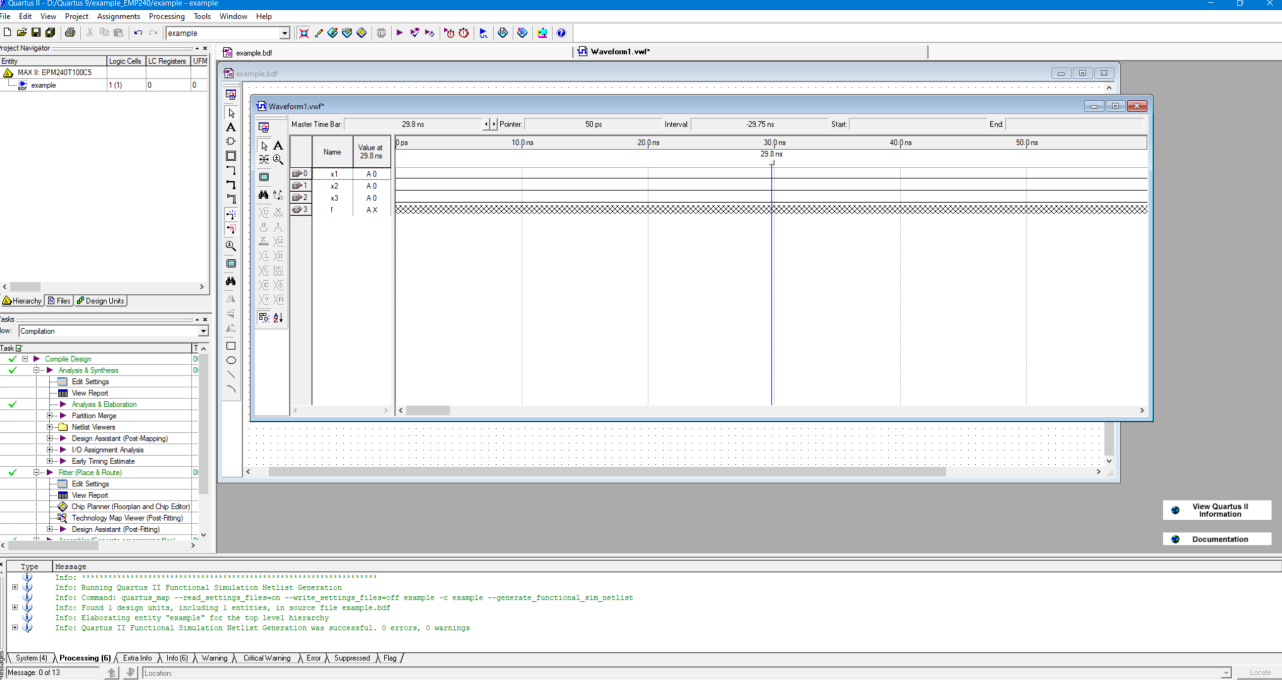
Далее выполняем пункт: File->New->Vector Waveform File.

В поле Name правой кнопкой мышки вызываем контекстное меню и выбираем в нем вкладку Insert->Insert Node or Bus….

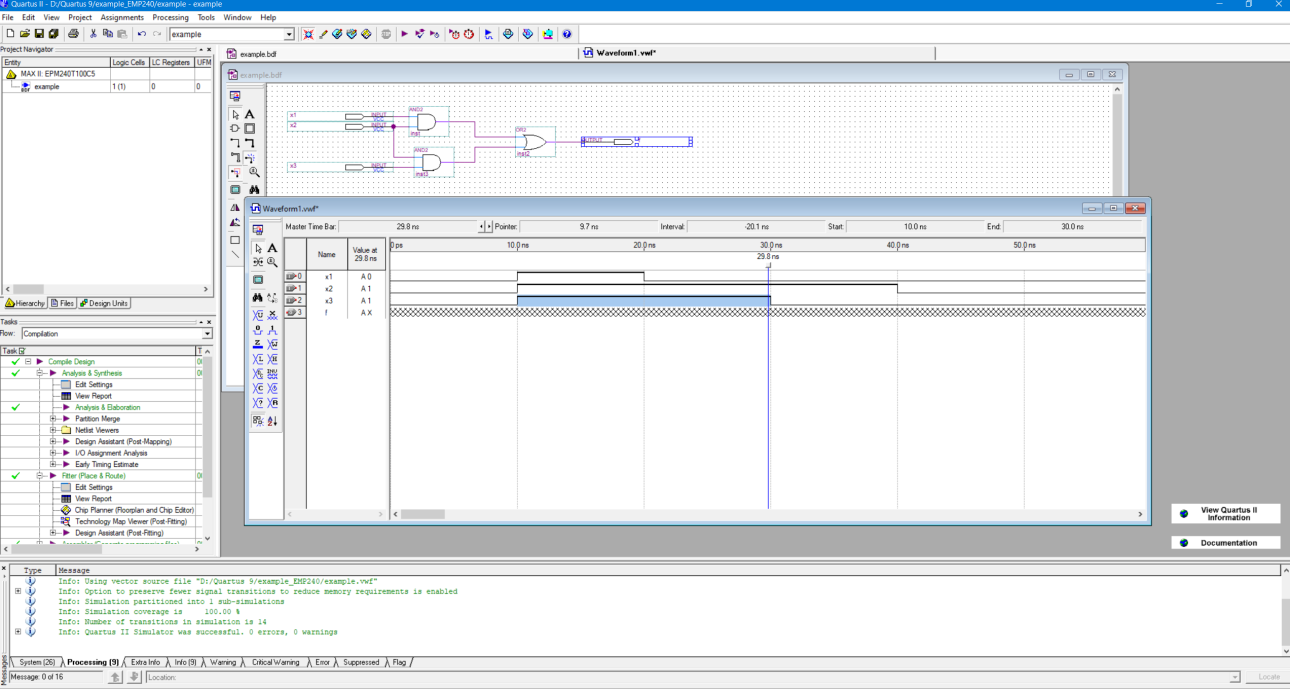
И далее нажимаем кнопку Node Finder…. Выбираем в окне Filter параметр аll и затем нажимаем кнопку List.

В выпавшем списке (переменных и функций) из поля Nodes Found перетаскиваем в поле Selected Node те из переменных и функций для которых необходимо выполнить моделирование. Можно изменять порядок их следования перетаскивая по одной (>) или перенести сразу все используя (>>). Например:

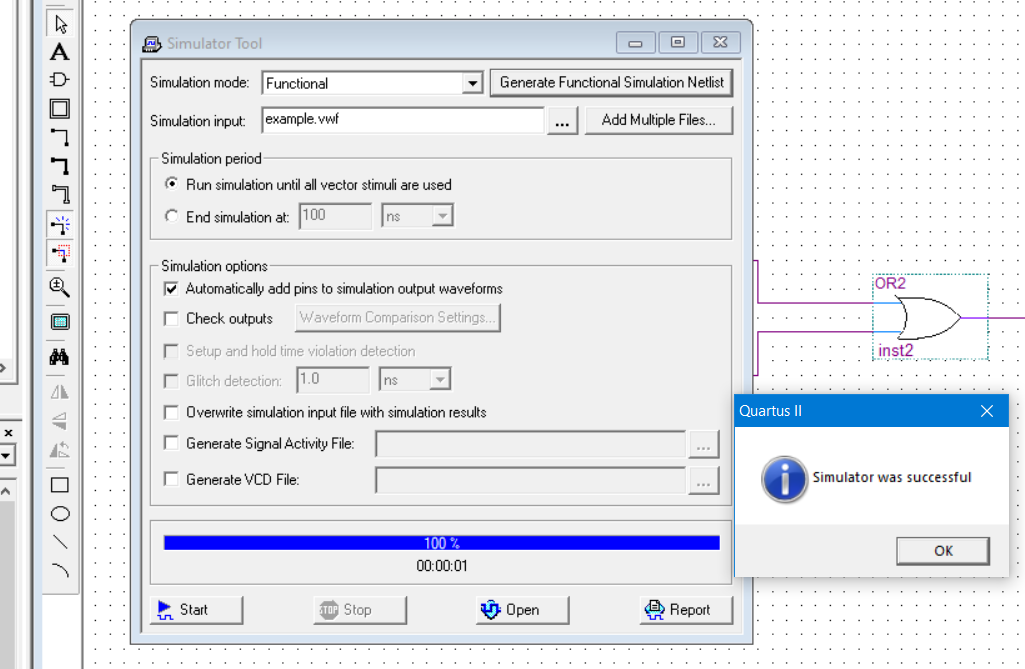


Нажав Ok получим:

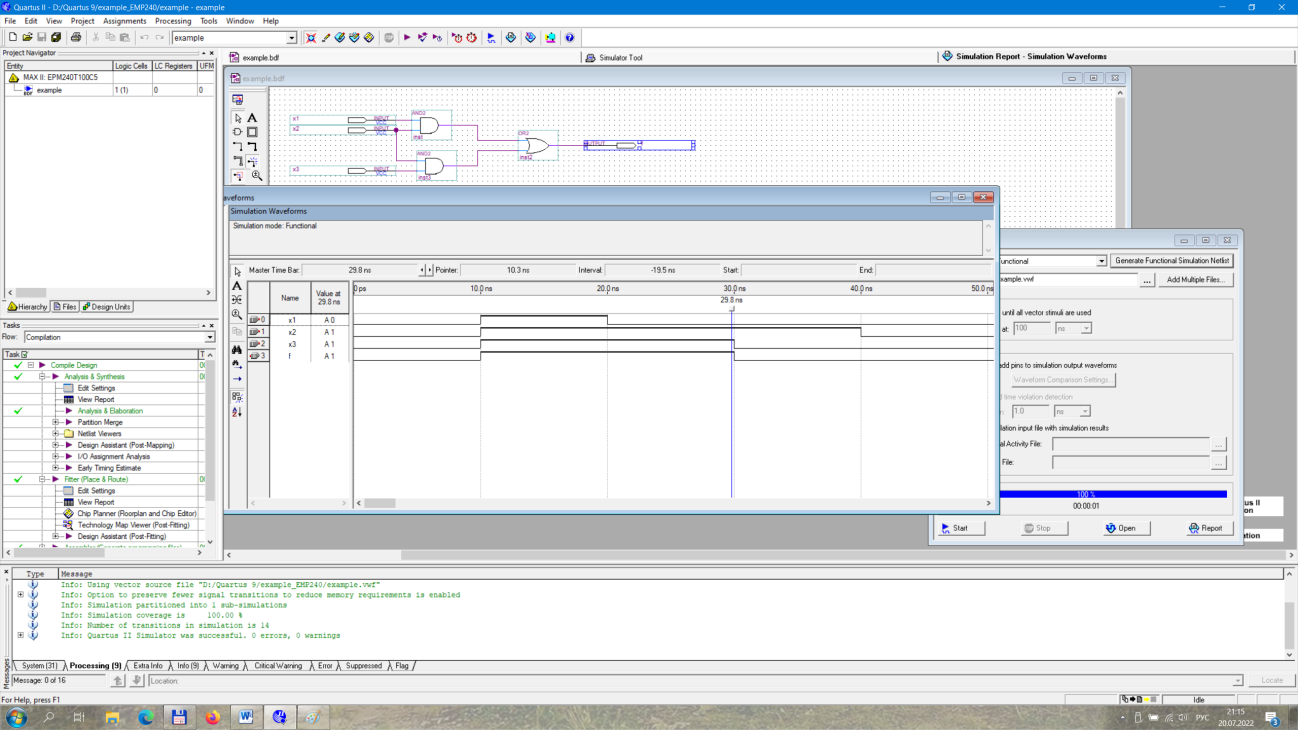
Для каждой из переменных (x1, x2, x3) выбираем интервал и устанавливаем для него значение переменной на этом интервале. Значение выбирается с помощью меню расположенного в окне слева.

Выполняем сохранение очередного этапа работы над проектом: File->Save.

Далее выполняем пункт: Processing->Simulator Tool. В выпавшем окне в пункте Simulation mode выбираем вкладку Functional (функциональное моделирование) или Timing (моделирование с учетом временных задержек) и нажимаем кнопку Generate Functional Simulation Netlist.



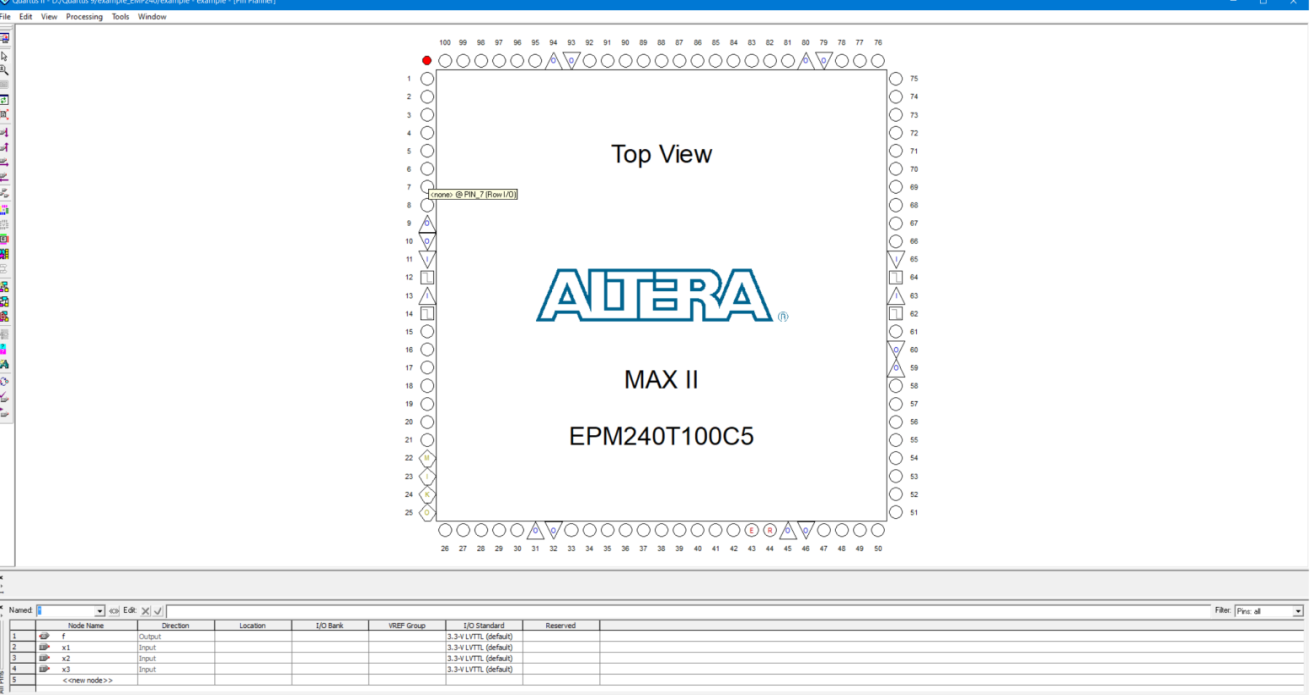
Далее нажимаем на кнопку Start и затем на кнопку Report. Получаем результат функционального моделирования работы схемы проекта



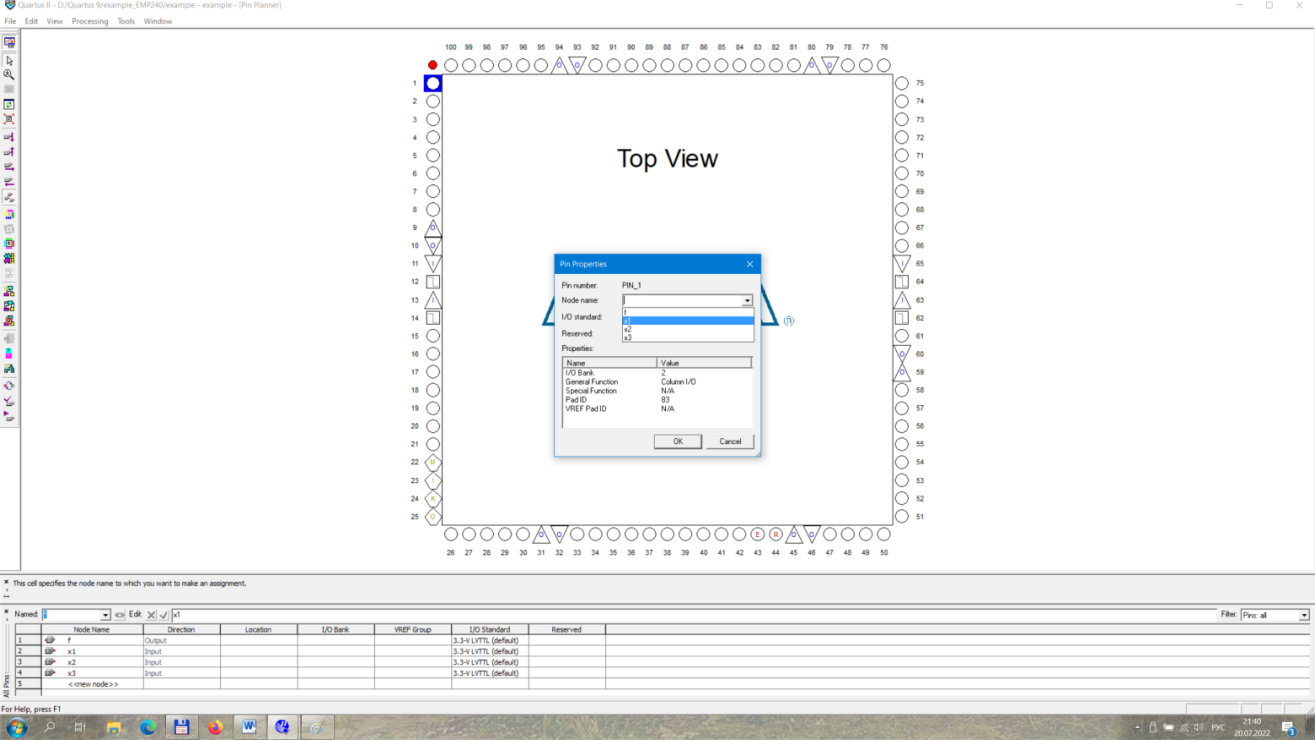
**Программирование и конфигурация (Programming and Configuration) проекта**

Все выполненное выше позволило создать схему и промоделировать ее работу. Но пока проект находится в памяти машины, на которой выполнялось создание проекта. Для переноса проекта (разработанной схемы) в память ПЛИС (Программируемая Логическая Интегральная Схема) выполняем следующие действия.

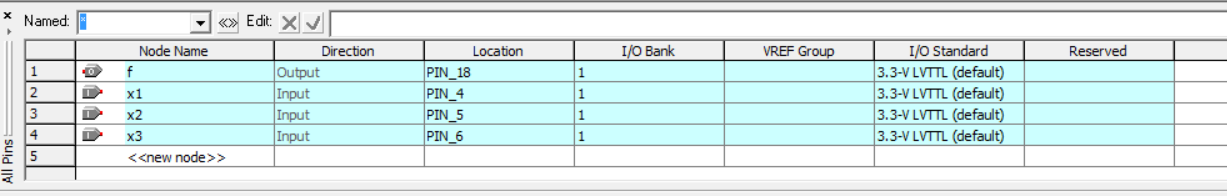
Выполняем пункт меню: Assignments->Pin Planner. На экран выводится информация о расположении и функциональном назначении pin микросхемы ПЛИС, для которой выполнялся проект (она выбрана была в самом начале создания проекта). При использовании отладочной платы нас интересует, какой pin платы (его номер) связан с каким pin микросхемы. Легко заметить, что рядом с pin микросхемы стоят номера (1-100). Это и есть номера pin платы, на которой расположена выбранная для проекта ПЛИС. Назначение каждого из pin можно получить наведя курсор на него.



Для того, чтобы «привязать» (назначить) переменную (функцию) из проекта к некоторому pin микросхемы (платы) необходимо двойным щелчком левой кнопки мыши вызвать контекстное меню. В поле Node Name в выпадающем меню выбрать некоторую переменную (функцию) назначаемую на этот pin. Так же в поле I/O Standart можно выбрать рабочее напряжение на этом pin.



Настройку (редактирование) выводов можно выполнить в окне «All pins»

Окно «All pins» оформлено в виде таблицы из семи колонок: «Node Name», «Direction», «Location» и «I/O Bank», «I/O Standart», «Reserved», «Current Strength».

Ячейки этой таблицы редактируются как привычным двойным кликом левой кнопки мыши, так и с помощью специализированного поля «Edit:» в верхней части окна «All pins».

Ячейки в колонках «Node Name» и «Location» допускают ввод новых значений с клавиатуры. В остальных колонках реализован выбор значений из выпадающего списка.

Рассмотрим назначение каждого столбца окна «All pins».

В первой колонке – «Node Name» перечисляются входы и выходы спроектированной схемы. Все существующие выводы схемы добавляются сюда автоматически после выполнения «Analysis & Synthesis». Список выводов в «Node Name» автоматически обновляется и при выполнении каждой новой компиляции. Следовательно, можно отредактировать схему: добавить или удалить несколько выводов. Запустить анализ и синтез схемы. В результате, среда автоматически добавит новые или удалит отсутствующие на схеме выводы. Удалять выводы из списка можно и вручную, например, кнопкой «Del» на клавиатуре.

Добавлять входы и выходы можно вручную, просто выполнив двойной клик на ячейке с текстом <<new node>> и введя имя вывода с клавиатуры. Аналогично можно отредактировать названия уже добавленных выводов схемы. Но имя вывода в «Node Name» должно до буквы совпадать с его именем на схеме, иначе Quartus не сможет выполнить их сопоставление и во время компиляции обозначит этот вывод как неизвестный.

Вторая колонка – «Direction» заполняется системой автоматически после анализа и синтеза схемы. Она недоступна для редактирования и несёт информационную функцию. В этой колонке указывается тип вывода схемы: «Input» – вход, «Output» – выход, «Bidir» – двунаправленный вывод и «Unknown» – вывод тип которого не удалось определить.

Наличие вывода с типом «Unknown» – свидетельствует об ошибке проектирования, либо был удален контакт схемы, с которым этот вывод был связан, либо это неправильно настроенный зарезервированный вывод.

Следующая пара колонок: «Location» и «I/O Bank». Именно в них определяется принадлежность входов и выходов схемы конкретным ножкам микросхемы.

Столбец «Location» дает возможность задать назначение для входа/выхода схемы с точностью до конкретной ножки чипа. Новое значение «Location» можно как ввести с клавиатуры, так и выбрать из выпадающего списка. Вызов выпадающего списка появляется через двойной клик левой кнопки мышы по редактируемому полю. Пример выпадающего списка «Location» представлен на следующем рисунке.

В списке доступны следующие варианты:

* EDGE\_BOTTOM, EDGE\_LEFT, EDGE\_RIGHT, EDGE\_TOP: указывают что настраиваемый вывод схемы будет автоматически выбран из выводов на нижней, левой, правой или верхней грани корпуса чипа соответственно.
* I/O BANK 1, I/O BANK 2: настраиваемый вывод схемы будет автоматически выбран из первого или второго банка выводов.
* PIN\_номер: назначение выводу схемы конкретной ножки микросхемы.

Первые две группы вариантов не определяют конкретной ножки микросхемы. Здесь выбор ножки остается за Quartus. Вывод микросхемы будет назначен средой автоматически при выполнении полной компиляции на этапе «Fitter (Place & Route)». Данные настройки предназначены для работы с широкими сигнальными шинами, когда множество дорожек должны идти от чипа рядом друг с другом. Параметр «EDGE\_…» является более общим по сравнению с «I/O BANK X», так как по одному краю корпуса чипа могут идти выводы разных банков.

Группировка же выводов в банки обуславливается внутренним устройством микросхемы Max II. Внутри чипа предусмотрены цепи для преобразования входного напряжения питания в различные по уровню выходные напряжения логических единицы и ноля. В EPM240T100C5 (EPM570T100C5) таких преобразователей всего два. Поэтому все доступные пользователю выводы разделены на две группы. Рабочее напряжение любой ножки в каждой из групп может настраиваться в широких пределах от 1,5 до 3,3 В. Но это напряжение должно быть одинаковым для всех ножек в пределах одного банка.

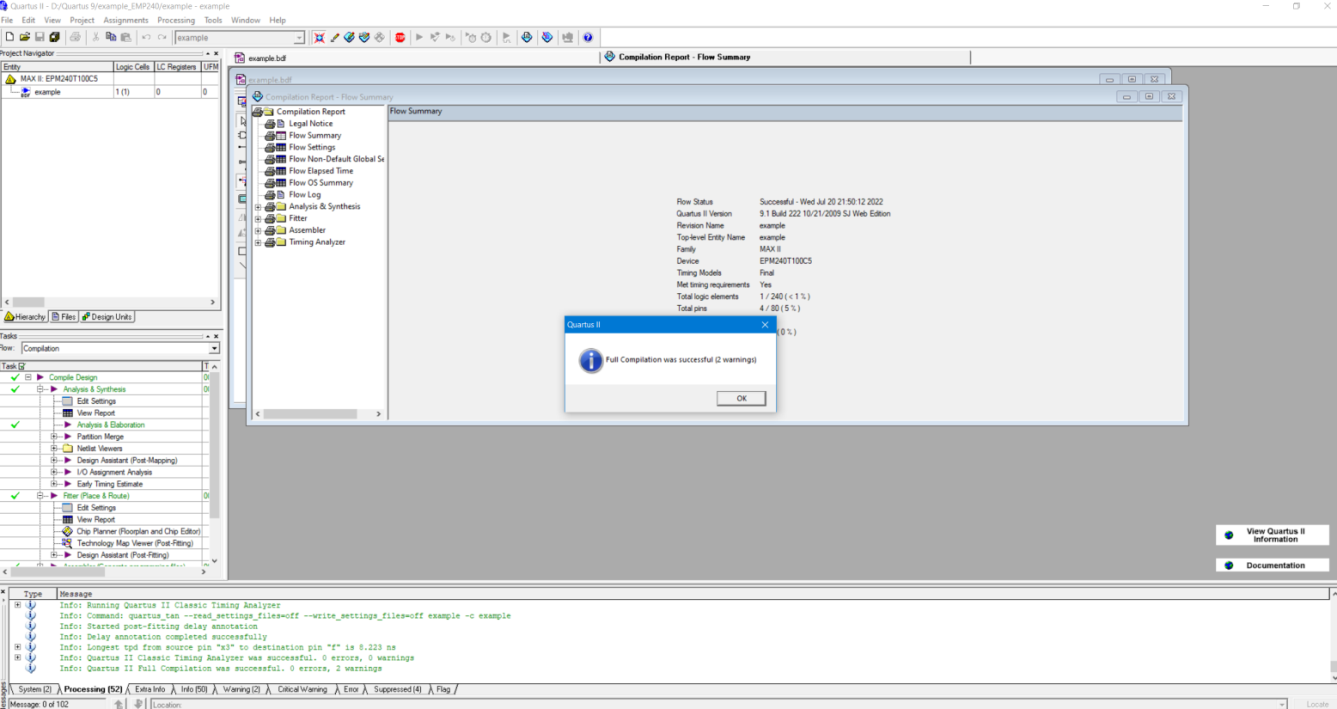
Для схем с небольшим числом входов и выходов наиболее удобным вариантом настройки является прямое указание конкретной ножки микросхемы — третий вариант. Эта настройка может быть выполнена как прямой правкой значения в «Location», так и простым перетаскиванием конкретного вывода схемы из столбца «Node Name» на требуемую ножку на изображении корпуса микросхемы.

Следующая колонка — «I/O Bank» позволяет указать банк выводов микросхемы. Этот параметр является взаимосвязанным с параметром «Location». При указании номера банка в колонке «I/O Bank», в «Location» будет автоматически подставлен соответствующий банк ввода‑вывода. И наоборот, при выборе банка или вывода в «Location» значение банка будет определено автоматически. Так как для малых проектов нет необходимости работать с банками ввода‑вывода, то подойдет и автоматически подставляемое значение «I/O Bank».

А вот следующий параметр «I/O Standart» является важным для любого проекта. Здесь настраивается стандарт ввода‑вывода для выбранной ножки микросхемы. Стандарт ввода‑вывода определяется его электрическими характеристиками, в первую очередь — это максимально допустимые уровни напряжения и тока. Как уж говорилось выше, ПЛИС Max II может работать в широком диапазоне напряжений, и очень важно выбрать правильное напряжение для своей схемы. Если напряжение сигнала приходящего на вход ПЛИС окажется выше заданного, то оно может вывести из строя как отдельную ножку ПЛИС, так и ядро микросхемы в целом. И наоборот, чрезмерно высокое выходное напряжение на ножке ПЛИС может вывести из строя подключенное к ней устройство.

Последняя из рассматриваемых — колонка «Reserved», она полезна для больших проектов. Зарезервированные выводы используются в случаях когда планируется расширение функционала спроектированного устройства. В этом случае можно развести плату. Подключить еще незадействованные выводы ПЛИС к схеме, и изготовить устройство не дожидаясь реализации всех запланированных возможностей. А позже, просто обновить конфигурацию ПЛИС. Но если зарезервированные выводы не присоединены к схеме внутри ПЛИС, то какие сигналы будут на них? Именно сигнал и задается в колонке «Reserved». Вывод можно подключить к питанию («As output driving VCC»), к «земле» («As output driving ground»), настроить как неподключенный вход («As input tri-stated») и так далее.

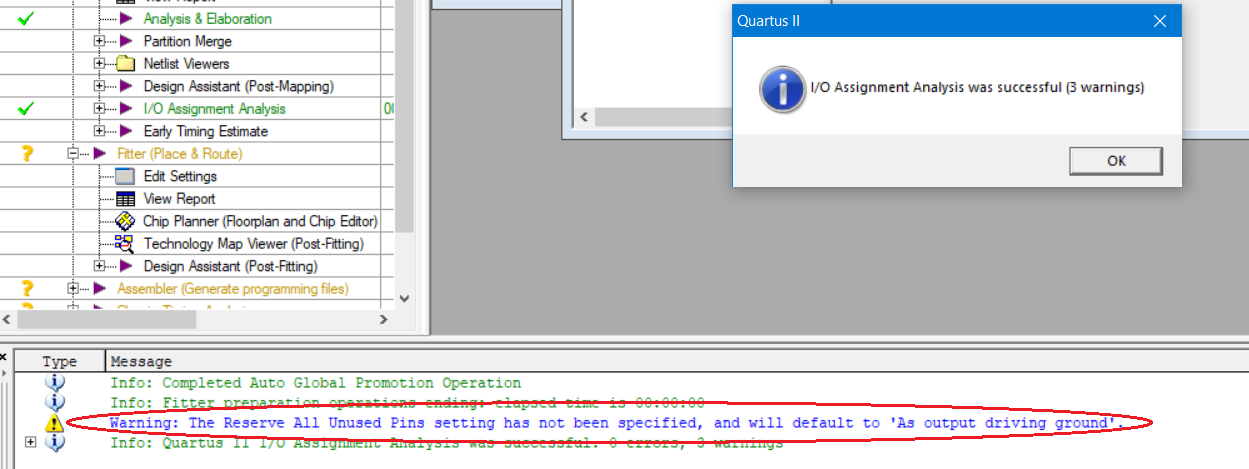
Значения параметров «I/O Standart» и «Current Strength» установленные по умолчанию: 3.3v LVTTL и 16mA соответственно, хорошо подходят для схемы. Параметры «Node Name» и «Direction» определяются системой автоматически. «Node Name» обычно не нуждается в ручной правке, а «Direction» не предоставляет такой возможности. И так как проект довольно мал, то производить назначения контактов микросхемы целыми банками или использовать зарезервированные выводы не требуется. Таким образом, единственным параметром требующим ручной настройки остается — «Location».

Далее необходимо выполнить полную компиляцию проекта: Processing->Start Compilation.

**Проверка назначений ввода-вывода**

После того, как все необходимые выводы микросхемы настроены, нужно проверить корректность сделанных настроек. Проверка настроек выводов в «Pin Planner» выполняется во время процесса «I/O Assignment Analysis». Запуск анализа выполненных назначений ввода‑вывода осуществляется с помощью главного меню: Processing ->  Start -> Start I/O Assignment Analysis.

Даже несмотря на безупречную настройку всех выводов, будет получено предупреждение следующего содержания: «Warning: The Reserve All Unused Pins setting has not been specified, and will default to ‘As output driving ground’.» Из него следует, что в проекте остаются какие-то ненастроенные выводы, которым будет задан некий параметр по умолчанию.

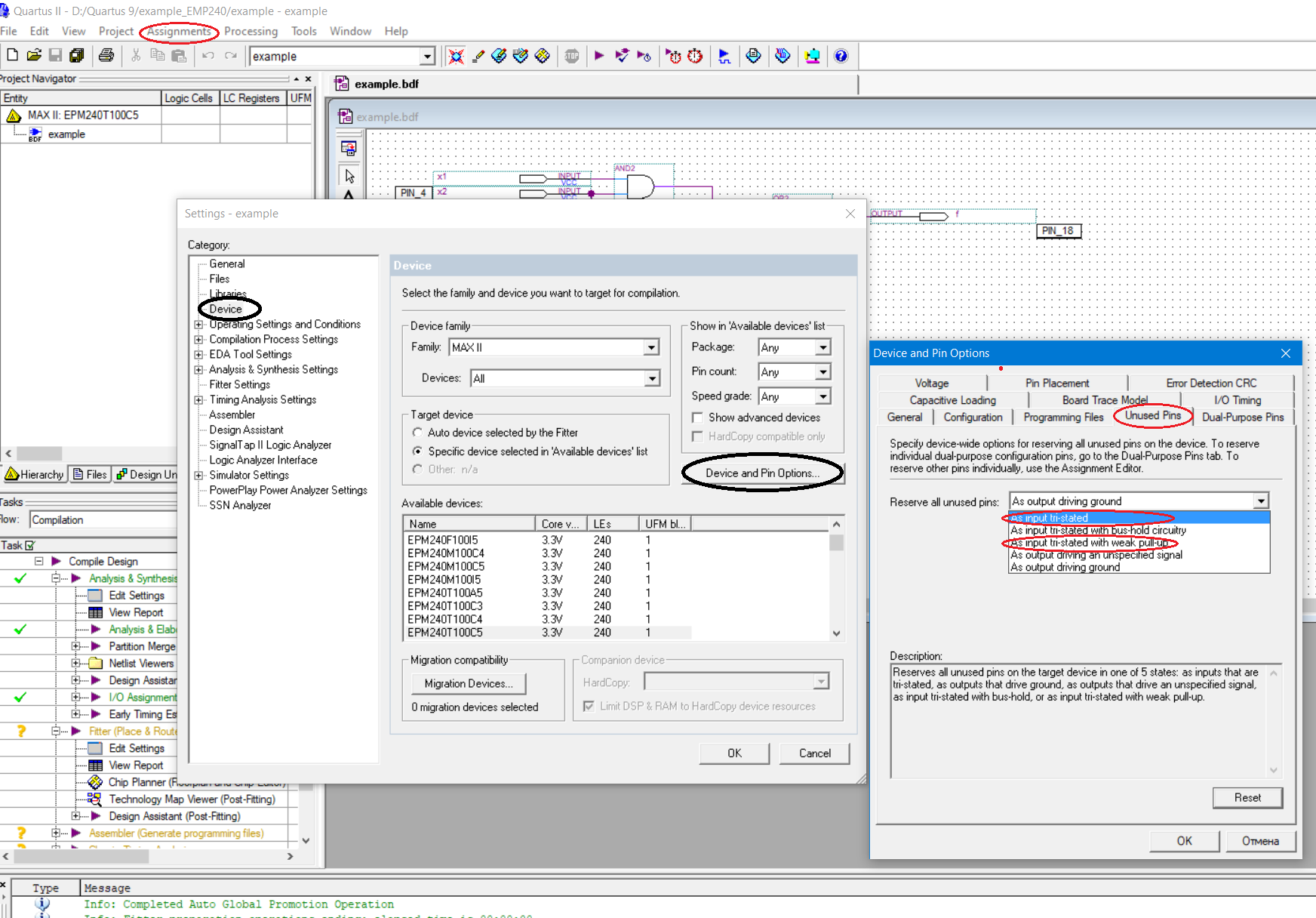


Речь идет обо всех контактах микросхемы, которые не используются в данном проекте. Настройка таких контактов выполняется не в «Pin Planner», а в параметрах проекта.

**Параметры проекта «Device and Pin Options»**

В проекте используется всего лишь 4 контакта из почти сотни доступных для пользователя контактов. Оставлять неподключенными входы микросхемы нежелательно (нельзя). И так как почти любой контакт ПЛИС можно настроить и как выход, и как вход, то необходимо обеспечить корректный уровень напряжения на каждом из них. ПЛИС Max II позволяет соединить любой из неиспользуемых контактов с плюсом или минусом питания с помощью своих внутренних цепей. Такое соединение настраивается с помощью параметра «Reserve all unused pins:».

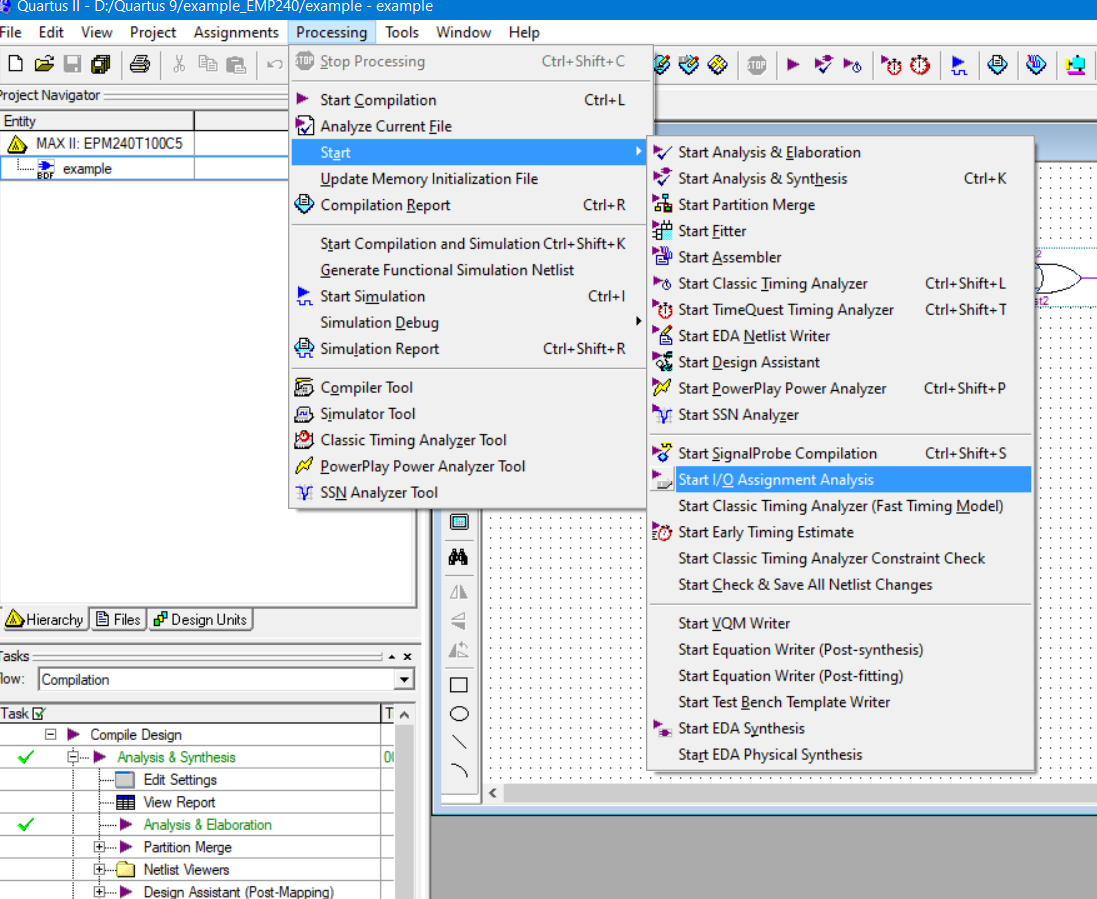
Для изменения настройки «Reserve all unused pins:» необходимо открыть окно «Device» (главное меню Quartus → Assignments → Device). В окне «Device» можно увидеть кнопку «Device and Pin Options…».

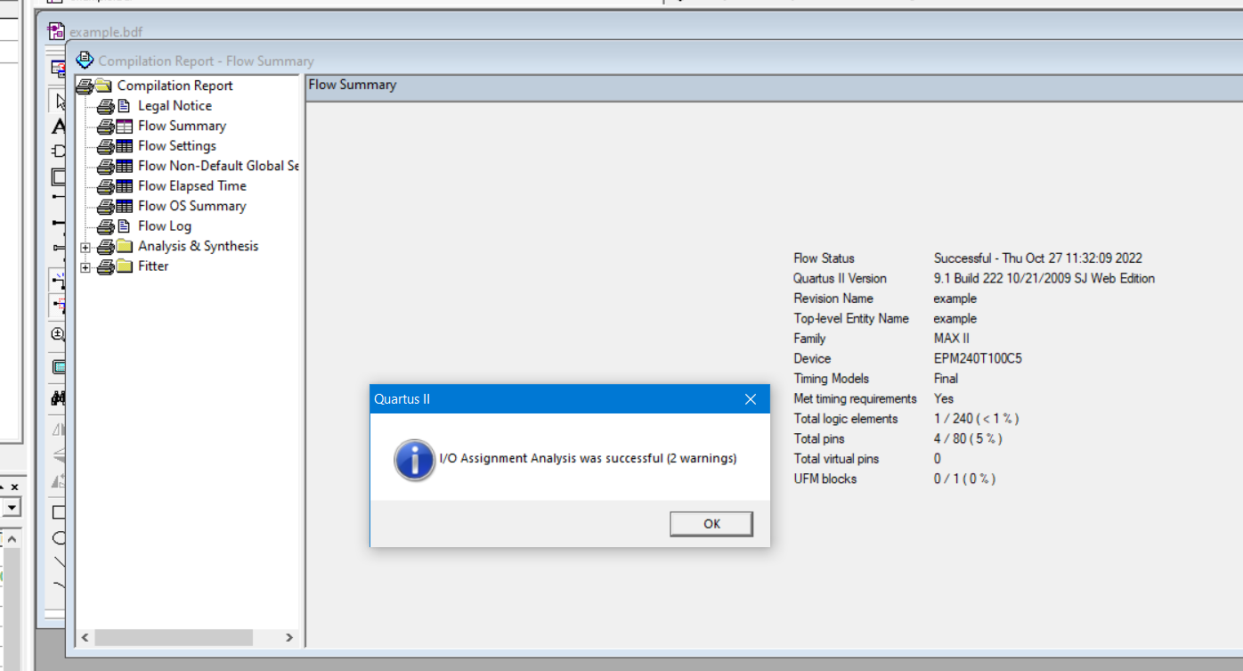


С помощью «Device and Pin Options…» вызывается окно «Device and Pin Options», в котором перечисляются параметры проекта: включение/выключение сигналов «DEV\_CLRn» и «DEV\_OE» (вкладка «General»), компиляция дополнительных файлов конфигурации (вкладка «Programming Files»), стандарт ввода‑вывода по умолчанию (вкладка «Voltage») и, среди прочего, вкладка «Unused pins» (выделенная на рисунке). Именно эта вкладка содержит параметр «Reserve all unused pins:».

По умолчанию параметр «Reserve all unused pins:» имеет значение «As output driving ground». Это означает, что все незадействованные в проекте контакты будут подключены к «земле» – минусу питания. Не самый удачный вариант. Ведь, если один из таких выводов окажется случайно соединен с плюсом питания, то получится короткое замыкание. Здесь лучше выбрать вариант «As input tri-stated» или «As input tri-stated with weak pull-up».

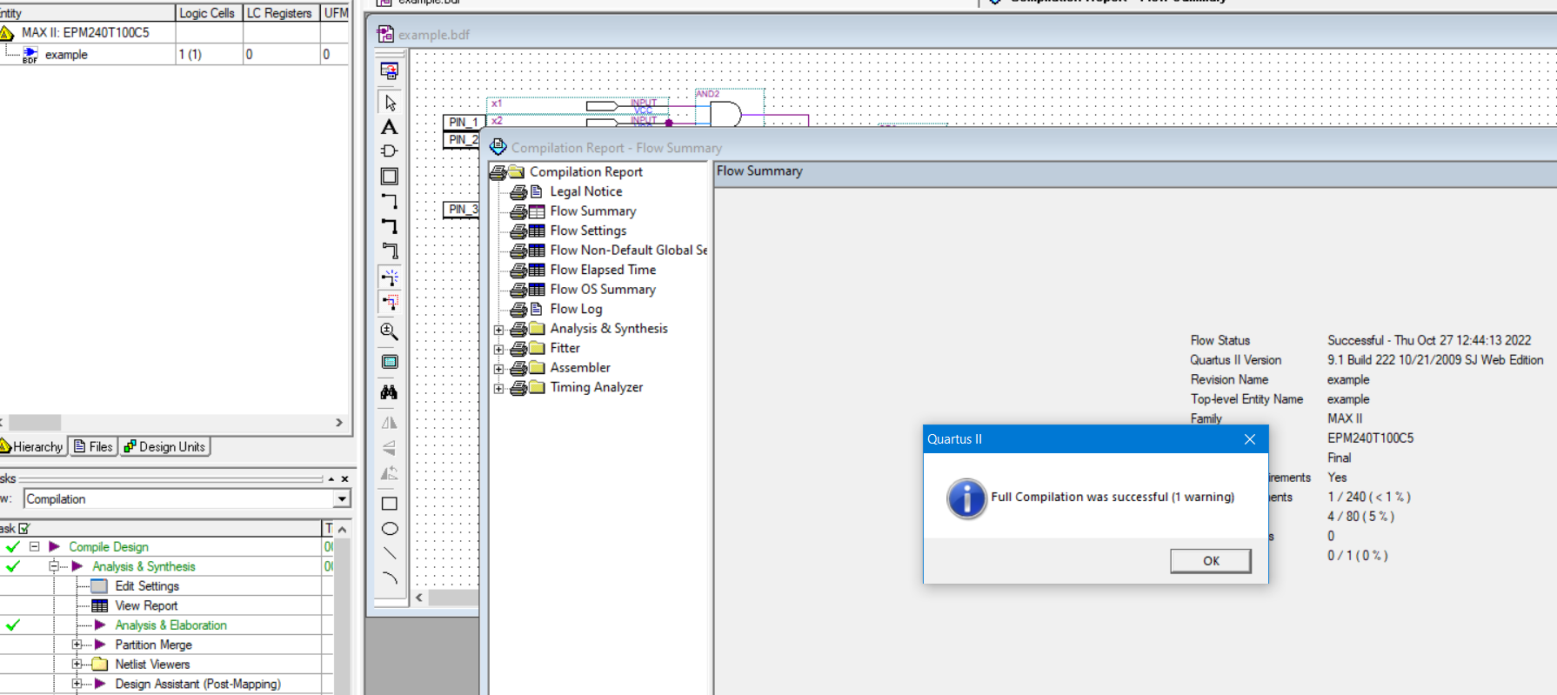
Выводы «As input tri-stated» переключаются в высокоимпендансное (High‑Z) состояние. В таком состоянии выводы практически полностью отключаются от внешних цепей, как если бы они не были подключены к ним физически. Недостаток такого режима работы в снижении помехоустойчивости схемы. Поэтому вариант «As input tri-stated with weak pull-up» оказывается более предпочтительным в большинстве случаев. При использовании вывода со слабой подтяжкой к питанию (weak pull‑up) в связанных с ним цепях гарантируется наличие напряжения высокого логического уровня, до тех пор, пока оно не будет изменено извне каким-либо подключенным устройством.

После настройки состояния неиспользуемых контактов еще раз выполните анализ сделанных назначений ввода‑вывода. Его можно запустить и из меню основного окна Quaturs: Processing → Start → I/O Assignment Analysis.



Предупреждение: «Warning: The Reserve All Unused Pins setting has not been specified, and will default to ‘As output driving ground’.» исчезло.

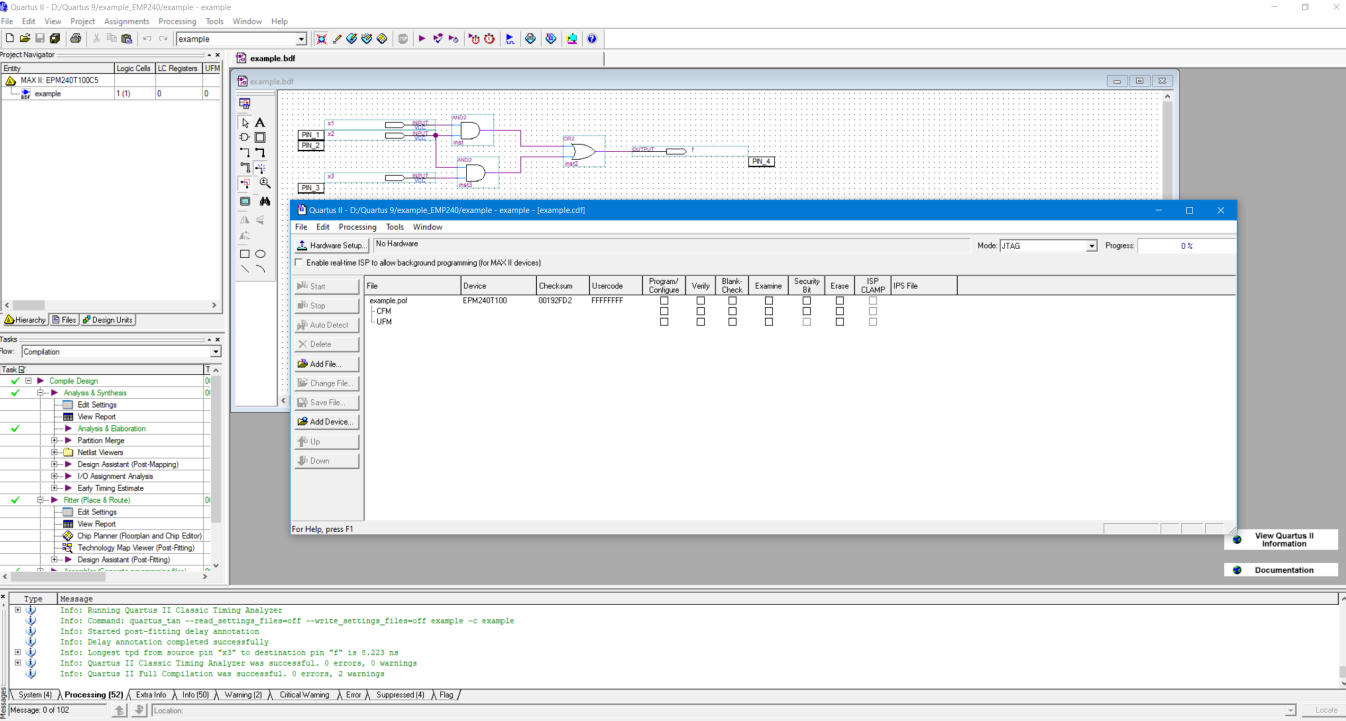
Настройка контактов ПЛИС успешно выполнена и можно переходить к заключительной части работы над проектом: его **полной компиляции** и загрузки скомпилированной конфигурации в ПЛИС.



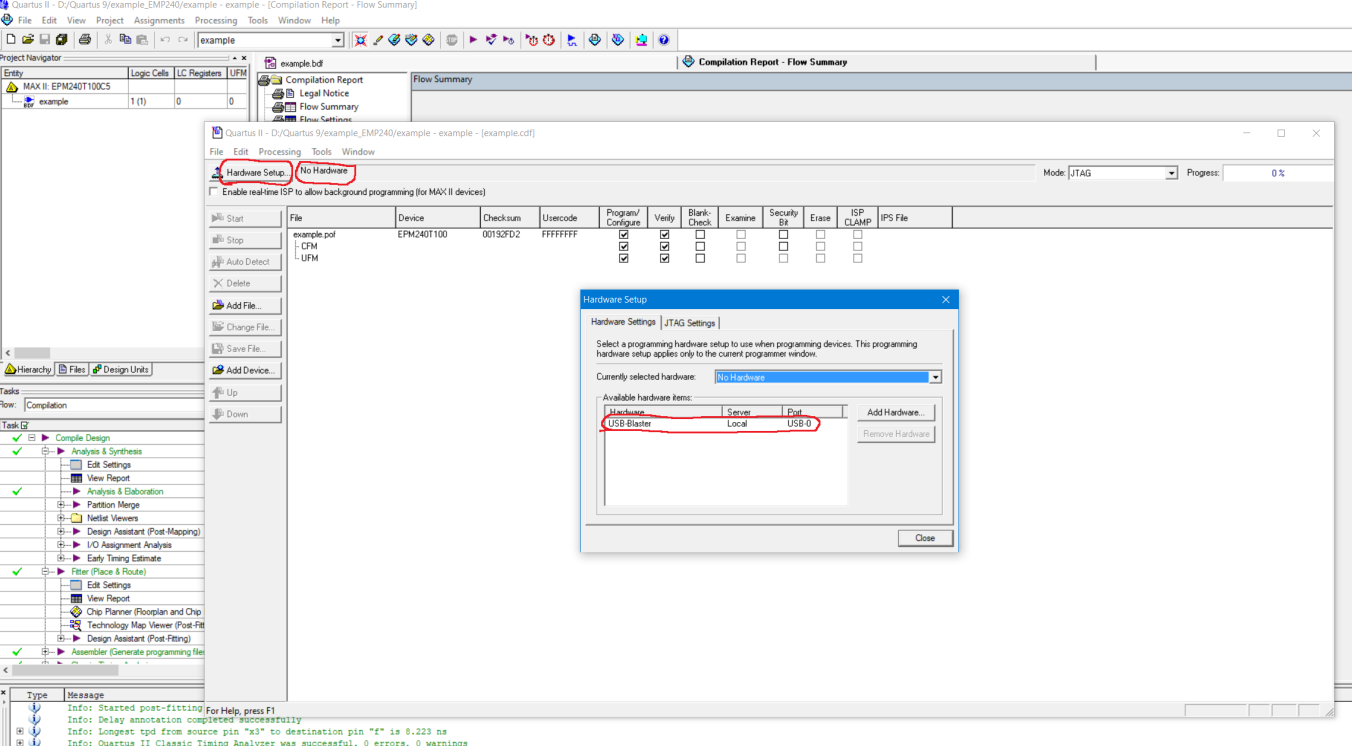
ПЛИС фирмы Altera поддерживают несколько режимов программирования. Рассмотрим загрузку прошивки через интерфейс JTAG. Подключаем программатор к разъёму JTAG на плате с ПЛИС.

После выполнения компиляции выбираем в пункт меню: Tools->Programmer.

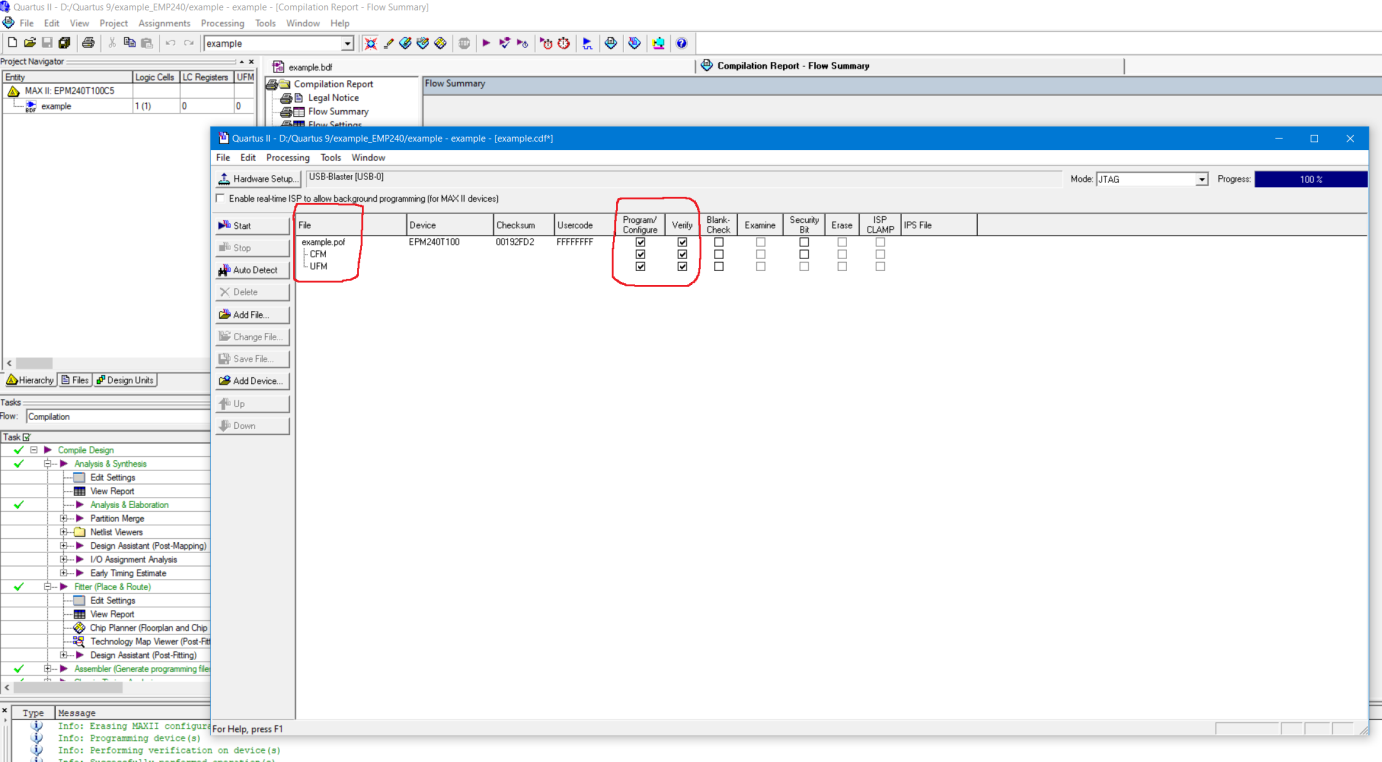
Если надпись на кнопке Start выделена, это означает, что программатор подключен к устройству и определен системой. В окне Programmer установить галочку Program/Configure и нажать кнопку Start. Прошивка будет записана в память ПЛИС. Процесс записи прошивки в ПЛИС отображается в окне Progress.

В противном случае, при переходе в окно программирования ПЛИС видно, что кнопка Start не подсвечена (надпись на ней не выделена жирным шрифтом). Выше видна вкладка с надписью No Hadware (отсутствует, либо не подключено устройство программирования ПЛИС – USB Blaster).

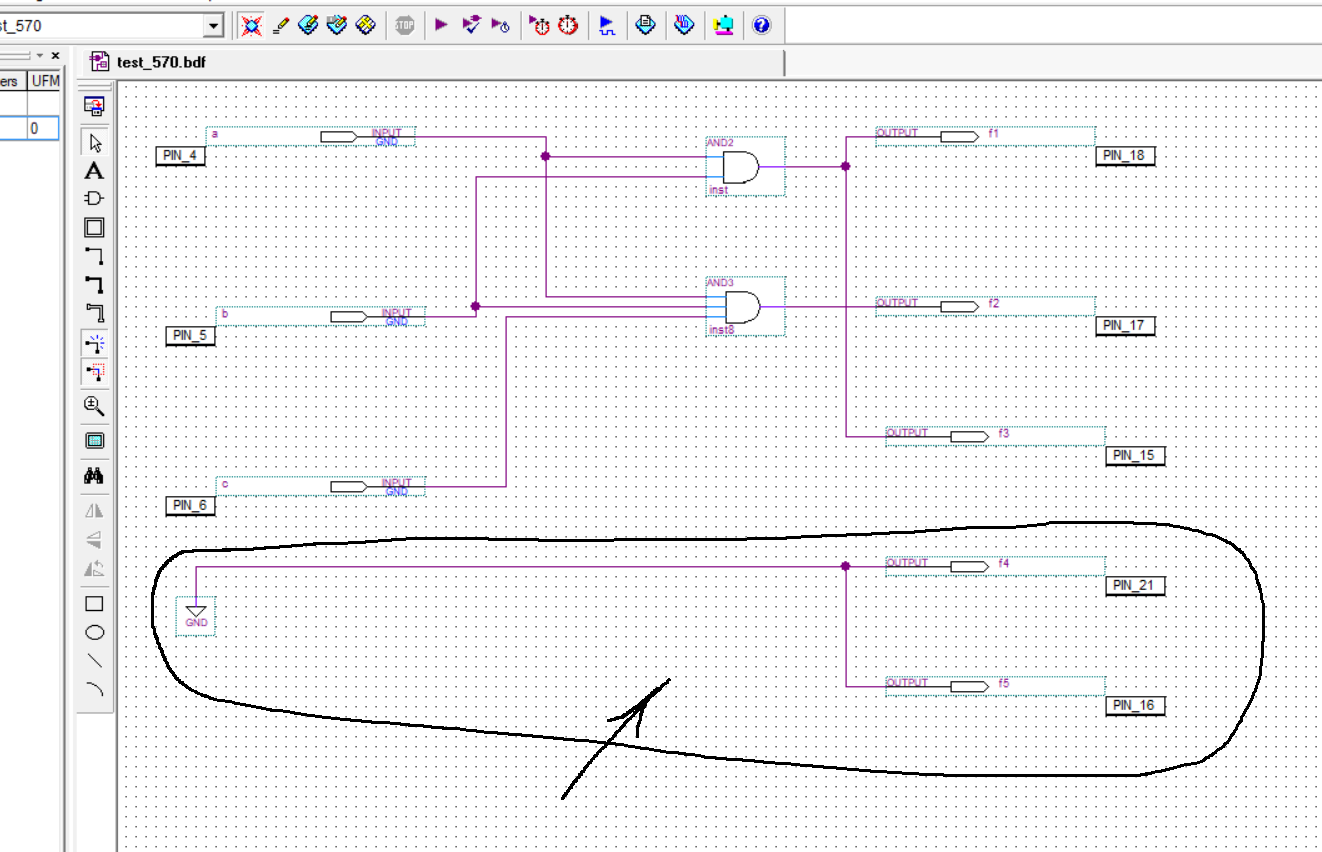
Добавим программатор для загрузки прошивки в выбранную ПЛИС. Для этого нажмём кнопку Hardware Setup**…** и в выпадающем списке выберем (двойным щелчком левой кнопки мыши) подключенный *USB Blaster*. Закроем окно *Hardware Setup*.



Если в столбце File отсутствует информация о проекте (файл \*.pof), то в окне программатора Programmer нажмите кнопку Auto Detect, чтобы Quartus попытался автоматически определить подключённую ПЛИС (если она не отображается) и файл прошивки. Файл прошивки создаётся Quartus по умолчанию при компиляции, если не задано иное. Далее, как и выше, в окне Programmer надо установить галочку Program/Configure и нажать кнопку Start. Прошивка будет записана в память ПЛИС. Процесс записи прошивки в ПЛИС отображается в окне Progress.



В заключение следует обратить внимание на то, что на ПЛИС могут быть выходы, соединенные с внешними элементами устройства, например, светодиодами, и при этом, некоторые из светодиодов в данном проекте не будут задействованы. При работе устройства на эти элементы схемы могут быть поданы с выходов ПЛИС произвольные значения. Это приведет к тому, что некоторые из светодиодов (не задействованные в проекте) могут быть включены произвольным образом. Чтобы исключить их произвольное включение можно поступить следующим образом. Данные выходы (pin) ПЛИС соединяются с «землей» (Gnd). Например, как это сделано на рисунке ниже.



В наличии имеются три устройства, разработанные на базе ПЛИС:

* устройсво 1 (6 входных переменных и 5 функций - для реализации простых устройств);
* устройсво 2 (для проверки работы устройств из курсовой работы (проекта));
* устройсво 3 (11 входных переменных и 8 функций - для мультиплексора, дешифратора и т.д.);

**Устройство 1 (для моделирования работы простых схем)**

Данное устройство позволяет моделировать работу системы не более чем 5 функций от 6 переменных, представленных в форме СДНФ. При создании схемы (ее моделирования на базе устройства 1) и ее дальнейшей проверки на устройстве важно учитывать следующее.

**Входные** переменные (х1 … х6) (их значения) подаются на следующие pin ПЛИС EPM240T100C5:

х1 - подается на pinIO 1 (номер pin разъема на плате ПЛИС)

х2 - подается на pinIO 2

х3 - подается на pinIO 3

х4 - подается на pinIO 4

х5 - подается на pinIO 5

х6 - подается на pinIO 6

**Выходные** функции (f1 … f5) (их значения) формируются (считываются) со следующих pin ПЛИС EPM240T100C5:

f1 - формируется на pinIO 33 (номер pin разъема на плате ПЛИС)

f2 - формируется на pinIO 34

f3 - формируется на pinIO 35

f4 - формируется на pinIO 36

f5 - формируется на pinIO 37

**Выходные** функции управления семисегментным индикатором (B, А, … C, ●) (их значения) формируются (считываются) со следующих pin ПЛИС EPM240T100C5:

B- формируется на pinIO 52 (номер pin разъема на плате ПЛИС)

A- формируется на pinIO 51

F- формируется на pinIO 48

A

G- формируется на pinIO 49

E- формируется на pinIO 44

G

F

B

D- формируется на pinIO 47

C- формируется на pinIO 42

C

E

●- формируется на pinIO 43

D

**.**

Общая схема внутреннего подключения для устройства 1:

разъем питания

ПЛИС

**5 v**

**плата ПЛИС**

3,3v

Gnd

Разъем JDEK ghjuhfvvте ПЛИС

Разъем на плате

36

34

37

35

33

4

2

1

3

5

кнопка вкл.

6

3,3v

Gnd

f5

f4

f3

f2

f1

x6

x5

x4

x3

x2

x1

разъем питания

разъем выходных функций

разъем входных переменных

светодиоды переменных x1  x2  x3  x4  x5   x6

светодиоды функций f1 f2 f3 f4 f5

51

49

50

47

Разъем на плате ПЛИС

C

D

G

A

B

`

48

F

E

**семисегментный индикатор**

Кнопки выбора x1 … x6

52

**устройство 1**

53

**.**

Входные переменные х1 – старшая … х6 – младшая (в таблице истинности)

Выходные функции f1 – старшая … f5 – младшая (в таблице истинности)

|  |  |  |
| --- | --- | --- |
| № | х1 х2 х3 х4 х5 х6 | f1 f2 f3 f4 f5 |
| 0 | 0 0 0 0 0 0 | 1 0 1 0 0 |
| 1 | 0 0 0 0 0 1 | 0 1 0 0 1 |
| 2 | 0 0 0 0 1 0 | х х х х х |
|  | **. . .** | **. . .** |
| 62 | 1 1 1 1 1 0 | 0 0 0 0 1 |
| 63 | 1 1 1 1 1 1 | х х х х х |

В таблице истинности в столбце f1 f2 f3 f4 f5 значения функций приведены просто для примера, в вашем случае они конечно будут иметь другие значения.

Для проверки правильности работы схемы на некотором наборе переменных x1…x6 необходимо, согласно таблицы истинности, используя (нажимая) кнопки устройства включить/выключить светодиоды, соответствующие переменным x1…x6 данного набора (строки) таблицы. При этом, если светодиод с номером i светится – переменная хi принимает значение 1, не светится – переменная хi равна 0. По реакции схемы (включении/выключении светодиодов соответствующих функциям f1…f5) можно судить о правильности (или ошибочности) работы схемы на данном (некотором) наборе входных переменных (выбранным из таблицы истинности).

**Устройство 2 (для курсовой работы по дисциплине «А и ЛО ЦУ»)**

При создании схемы для курсовой работы и ее дальнейшей проверки на устройстве важно учитывать следующее.

**Входные** переменные (х1 … х6) (их значения) подаются на следующие pin ПЛИС EPM240T100C5 (EPM570T100C5):

х1 - подается на pinIO 1 (номер pin разъема на плате ПЛИС)

х2 - подается на pinIO 2

х3 - подается на pinIO 3

х4 - подается на pinIO 4

х5 - подается на pinIO 5

х6 - подается на pinIO 6

**Выходные** функции (f1 … f5) (их значения) формируются (считываются) со следующих pin ПЛИС:

для EPM240T100C5 № pin для EPM570T100C5 № pin

f1 - формируется на pinIO 49 f1 - формируется на pinIO 18

f2 - формируется на pinIO 51 f2 - формируется на pinIO 17

f3 - формируется на pinIO 50 f3 - формируется на pinIO 15

f4 - формируется на pinIO 53 f4 - формируется на pinIO 16

f5 - формируется на pinIO 52 f5 - формируется на pinIO 21

Данное устройство позволяет выполнить проверку правильности работы построенной схемы одного из устройств (ОЧУ, ОЧУС, ОЧС) по исходной таблице истинности составленной в ходе выполнения курсовой работы (проекта). Устройство 2 изображено на рисунке.

Разъем JDEK ghjuhfvvте ПЛИС

`

светодиоды функций f1 f2 f3 f4 f5

светодиоды переменных x1  x2  x3  x4  x5   x6

Приемник SD карты

кнопка

SDcard

TFT LCD

разъем

питания

устройства

**7–9 v**

**устройство 2**

Для «прошивки» (занесения синтезированной и построенной в Quartus схемы в ПЛИС) можно поступить несколькими способами. Вначале подключить JDEK программатор к устройству.

Далее, в первом случае, не вставляя SD карту в устройство, включить его (подать на него питание). На экран будет выведено сообщение:

ERROR

not found SD-card

programming only CPLD

В нем говорится о том, что возможен (при отсутствии в устройстве SD карты) только режим занесения схемы в ПЛИС. В среде Quartus II, выполнив компиляцию своего проекта и убедившись, что нет ошибок, выбираем в меню: Tools->Programmer и переносим схему в ПЛИС, находящуюся в устройстве 2. Далее, вставляем SD карту в устройство и выполняем проверку схемы, нажав на кнопку.

Во втором случае (более универсальном), вставив SD карту и включив устройство, на экран будет выведено сообщение:

    -- to check the project,

    click on the botton

    -- otherwise GPLD/FPGA

    programming mode

    enabled

В нем сообщается, что есть два режима работы: программирование и контроль работы схемы. Используя JDEK программатор можно заносить схему в ПЛИС и затем, нажав на кнопку, ее проверять. В этом случае, если возникли ошибки в работе схемы, ее можно подкорректировать в Quartus (не отключая устройства и не извлекая SD карту) и вновь перенести ее в ПЛИС. Затем повторить проверку схемы.

Ниже приведена структура содержимого файла file.TXT с информацией для проверки правильности функционирования разработанной схемы.

В первой строчке файла содержится название устройства:

* PM\_DP\_KOD – преобразователь множителя в дополнительном коде;
* PM\_PR\_KOD – преобразователь множителя в прямом коде;
* OCHS – ОЧС (одноразрядный четверичный сумматор)
* OCHUS – ОЧУС (одноразрядный четверичный сумматор-умножитель)

Во второй строке приведена информация о входных переменных и выходных функциях.

В третьей и последующих строках содержится информация из таблицы истинности того устройства, для которого выполнялся проект (разрабатывалась схема).



Необходимо в файле в соответствии со своей информацией (таблицей истинности) внести изменения в столбцы с выходной информацией (функциями).

**Замечание:** при корректировке файла использовать режим замещения (а не вставки). Т.е. не нарушать структуру файла. Так же можно (но не обязательно) изменить имена входных переменных и выходных функций на свои (из своей таблицы истинности).

**Устройство 3 (для моделирования работы мультиплексора, дешифратора и т.д.)**

Данное устройство, реализованное на ПЛИС EPM240T100C5, позволяет моделировать работу несколько более сложных (по сравнению с устройством 1) схем. С помощью этого устройства можно моделировать работу устройств, определяемых системой до 8 булевых функций от 11 входных переменных, представленных в форме СДНФ (или упрощенных до ДНФ). При создании схемы (ее моделирования на базе устройства 3) и ее дальнейшей проверки на устройстве важно учитывать следующее.

**Входные** переменные (х1 … х11) (их значения) подаются на следующие pin ПЛИС EPM240T100C5:

х1 - подается на pinIO 1 (номер pin разъема на плате ПЛИС)

х2 - подается на pinIO 2

х3 - подается на pinIO 3

х4 - подается на pinIO 4

х5 - подается на pinIO 5

х6 - подается на pinIO 6

х7 - подается на pinIO 76

х8 - подается на pinIO 74

х9 - подается на pinIO 72

х10 - подается на pinIO 70

х11 - подается на pinIO 68

**Выходные** функции (f1 … f8) (их значения) формируются (считываются) со следующих pin ПЛИС EPM240T100C5:

f1 - формируется на pinIO 18 (номер pin разъема на плате ПЛИС)

f2 - формируется на pinIO 17

f3 - формируется на pinIO 15

f4 - формируется на pinIO 16

f5 - формируется на pinIO 21

f6 - формируется на pinIO 58

f7 - формируется на pinIO 56

f8 - формируется на pinIO 54

Выходные функции управления семисегментным индикатором (B, A, F,…, C, ●) (их значения) формируются (считываются) со следующих pin ПЛИС EPM240T100C5:

B- формируется на pinIO 41 (номер pin разъема на плате ПЛИС)

A- формируется на pinIO 42

F- формируется на pinIO 39

A

G- формируется на pinIO 40

E- формируется на pinIO 37

G

F

B

D- формируется на pinIO 38

C- формируется на pinIO 35

C

E

●- формируется на pinIO 36

**.**

D

Общая схема устройства 3:

ПЛИС

разъем питания

ПЛИС

**5 v**

Разъем JDEK ghjuhfvvте ПЛИС

светодиоды функций f1 f2 f7 f8

светодиоды переменных x1  x2  . . . x10   x11

. . .

`

**семисегментный индикатор**

. . .

Кнопки выбора x1 … x11

**устройство 3**

Входные переменные х1 – старшая … х11 – младшая (в таблице истинности)

Выходные функции f1 – старшая … f8 – младшая (в таблице истинности)

|  |  |  |
| --- | --- | --- |
| № | х1 х2 х3 х4 х5 х6 х7 х8 х9 х10 х11 | f1 f2 f3 f4 f5 f6 f7 f8 |
| 0 | 0 0 0 0 0 0 0 0 0 0 0 | 1 0 1 0 0 1 0 1 |
| 1 | 0 0 0 0 0 0 0 0 0 0 1 | 0 1 0 0 1 0 0 0 |
| 2 | 0 0 0 0 0 0 0 0 0 1 0 | х х х х х х х х |
|  | **. . .** | **. . .** |
| 62 | 1 1 1 1 1 1 1 1 1 1 0 | 0 0 0 0 1 0 0 1 |
| 63 | 1 1 1 1 1 1 1 1 1 1 1 | х х х х х х х х |

Ниже, для наглядности, справочно, приведен план расположения контактов для некоторых ПЛИС.

