

Пакет ModelSim и пакет Quartus (варианты совместной работы)

Цели

Данная работа посвящена знакомству с тем, как:

- 1 Запустить пакет ModelSim из пакета Quartus.
- 2 Как использовать IP из библиотеки пакета Quartus при моделировании в пакете ModelSim.

Обзор проекта

Структура проекта приведена на Figure 1.

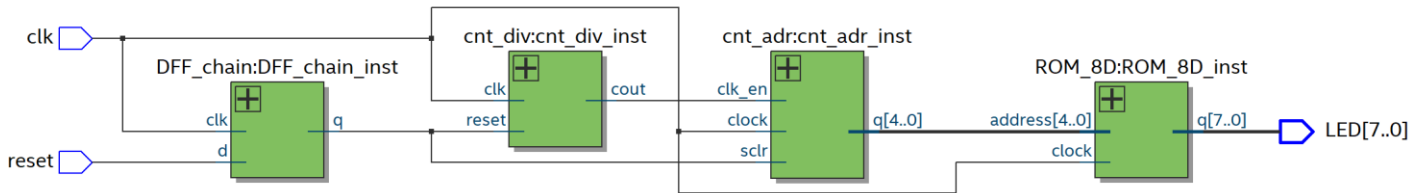


Figure 1

Входы проекта:

- clk – тактовый сигнал (частота 25MHz для платы MiniDilab-CIV).
- reset – сигнал сброса, поступающий от кнопки на плате MiniDilab-CIV.

Выходы проекта:

- LED[7:0] – выводы FPGA, соединенные со светодиодами на плате MiniDilab-CIV.

Алгоритм работы проекта:

- Модуль **cnt_div** – счетчик делитель, который делит входную частоту. Коэффициент деления – параметра модуля.
- Модуль **cnt_adr** – счетчик адреса, который формирует адреса для ROM памяти, реализованной как модуль **ROM_8D**.
- Модуль **ROM_8D** – модуль ROM памяти хранит 8-ми разрядные данные и в соответствии с текущим адресом формирует выходное слово, поступающее на выходы LED[7..0].
- Сигнал reset, активный уровень = 1, сбрасывает все счетчики в 0.
- Модуль **DFF_chain** – модуль синхронизации, который обеспечивает привязку асинхронного сигнала reset к тактовой частоте проекта. Модуль содержит два последовательно включенных триггера.

Создайте проект в пакете Quartus

- 1 Параметры проекта:
 - a. Рабочая папка: **C:/Intel_trn/Q_MS/lab_QMS1**
 - b. Project name: lab_QMS1
 - c. Top-Level design entity: lab_QMS1
 - d. Project Type: Empty project
 - e. Device: EP4CE6E22C8
 - f. **EDA tools: ModelSim Altera; Verilog HDL** (see Figure 2)

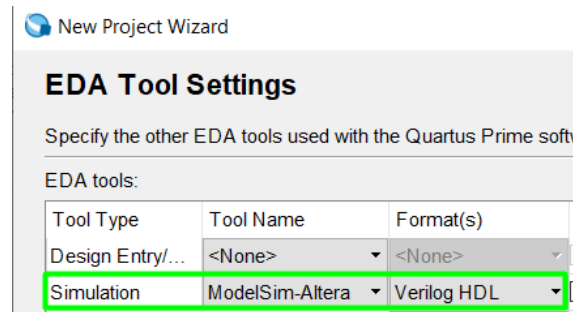


Figure 2

Добавьте IP модули к проекту

- 1 Создайте модуль **cnt_adr** используя LPM_COUNTER из библиотеки IP пакета Quartus.
 - a. Найдите модуль LPM_COUNTER в библиотеке IP Catalog (если библиотека IP Catalog не отображается – выполните команду View => Utility Windows => IP Catalog).

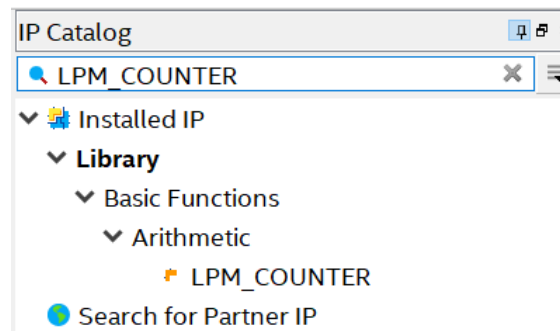


Figure 3

- b. Дважды щелкните модуль LPM_COUNTER.
 - c. Задайте имя **cnt_adr** в окне **Save IP Variation** и щелкните **OK**.

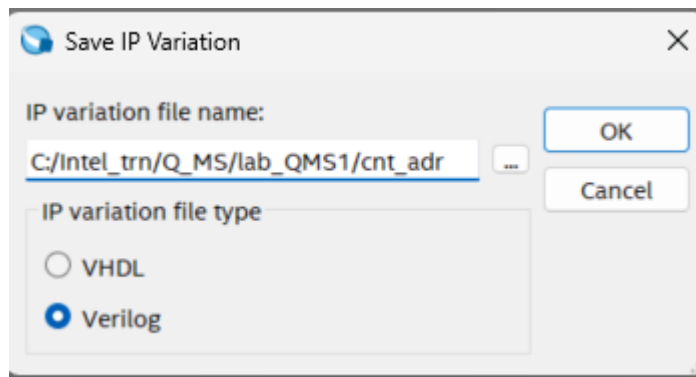


Figure 4

- d. Укажите разрядность **5 бит**, остальные настройки оставьте без изменения.

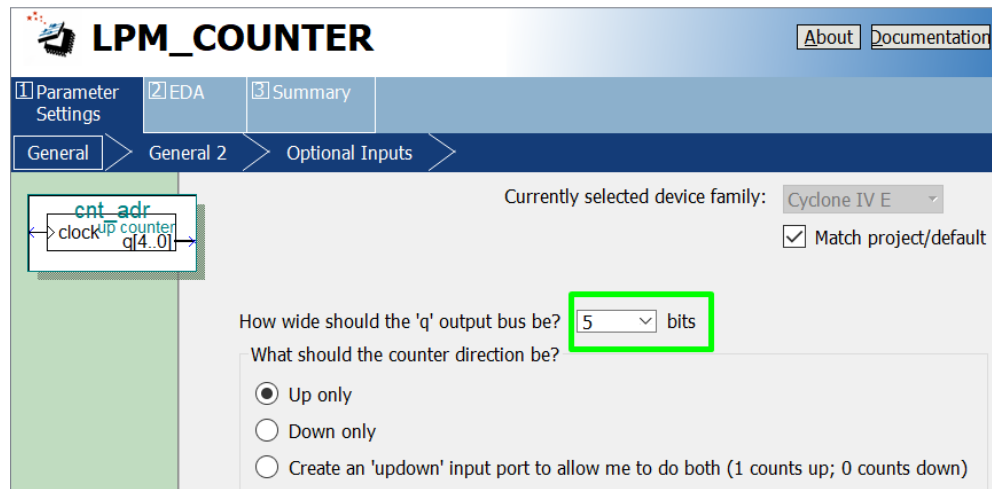


Figure 5

Щелкните **Next**.

- e. В появившемся окне установите **Clock enable**, остальные настройки оставьте без изменения.

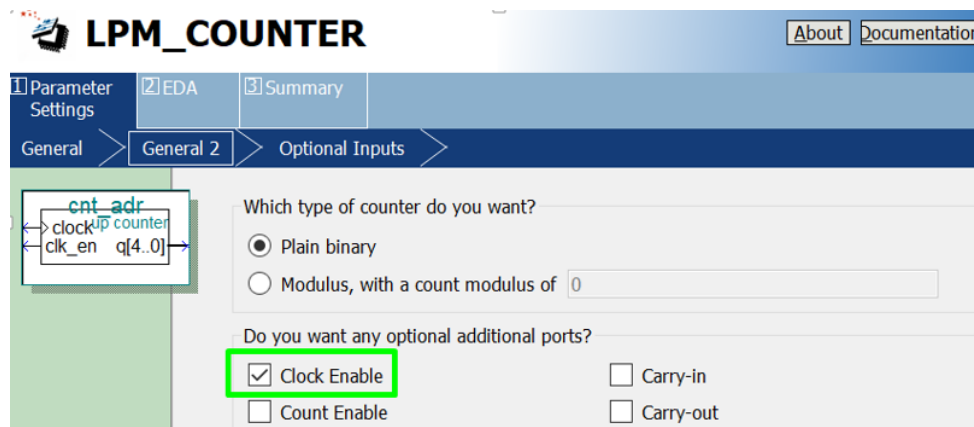


Figure 6

Щелкните **Next**.

- f. В появившемся окне выберите **synchronous Clear**, остальные настройки оставьте без изменения.

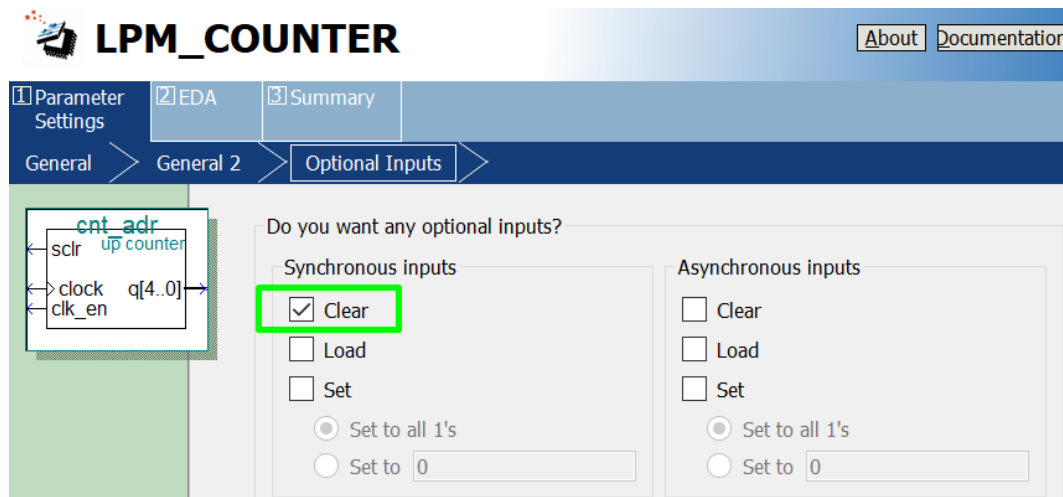


Figure 7

Щелкните **Next**.

g. В появившемся окне обратите внимание на указанную библиотеку, которая будет использована при моделировании. Эта библиотека называется **lpm** – библиотека моделей IP функций. Щелкните **Next**.

h. В появившемся окне выберите **cnt_adr_inst.v** и **cnt_adr_bb.v** – в рабочей папке появятся заготовки для использования модуля при иерархическом моделировании. Щелкните **Finish**.

i. Появится диалоговое окно, в котором предлагается добавить описание созданного IP модуля к проекту. Щелкните **Yes**. Файл **cnt_adr.qip** будет добавлен к проекту.

2 Создайте модуль **ROM_8D** используя IP компонент **ROM: 1-PORT** из IP библиотеки пакета Quartus.

Модуль **ROM_8D** хранит данные, заданные в файле **ROM_8D.hex**.

a. Найдите IP модуль **ROM: 1-PORT**.

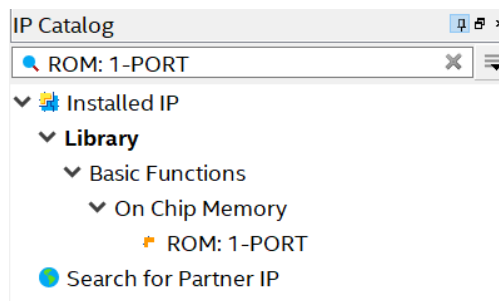


Figure 8

b. Дважды щелкните модуль **ROM: 1-PORT**.

c. Задайте имя **ROM_8D** в окне **Save IP Variation** и щелкните **OK**.

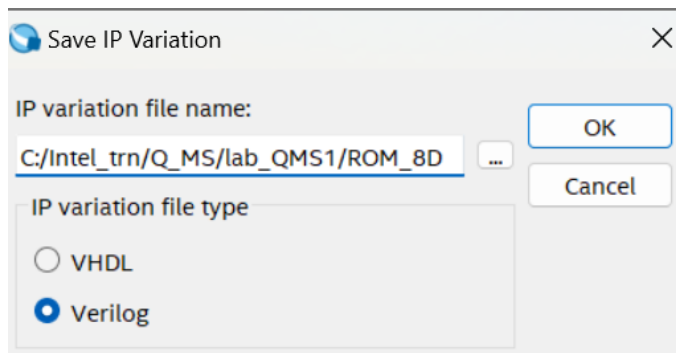


Figure 9

- d. Задайте - 32 слова по 8 бит каждое.

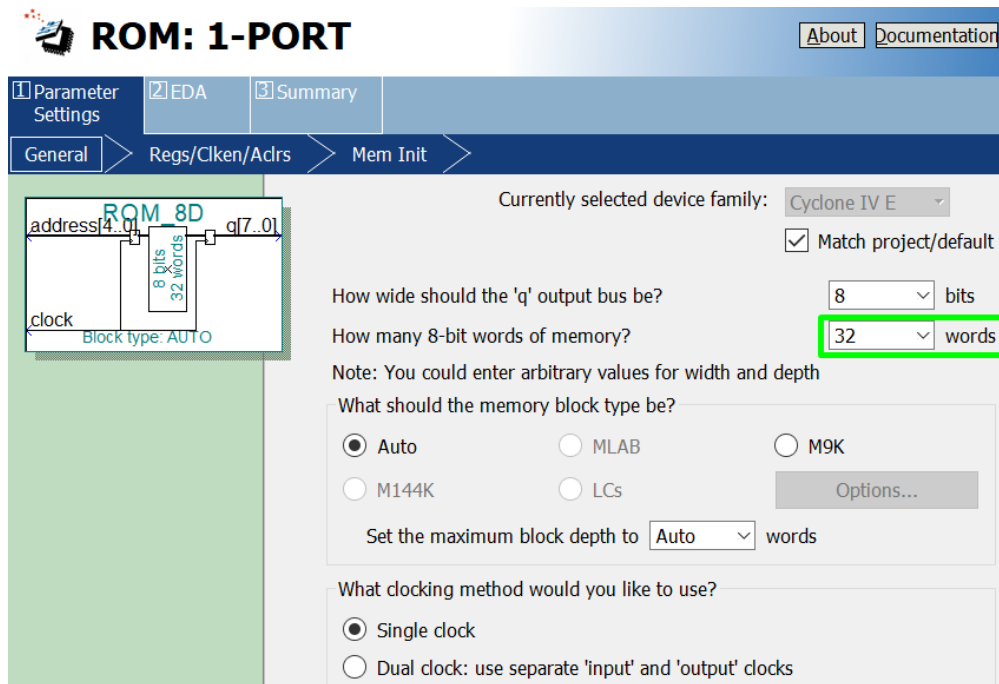


Figure 10

Щелкните **Next**.

- e. В появившемся окне щелкните **Next**.
 f. В появившемся окне укажите **ROM_8D.hex** как файл, содержащий исходные данные для модуля памяти.

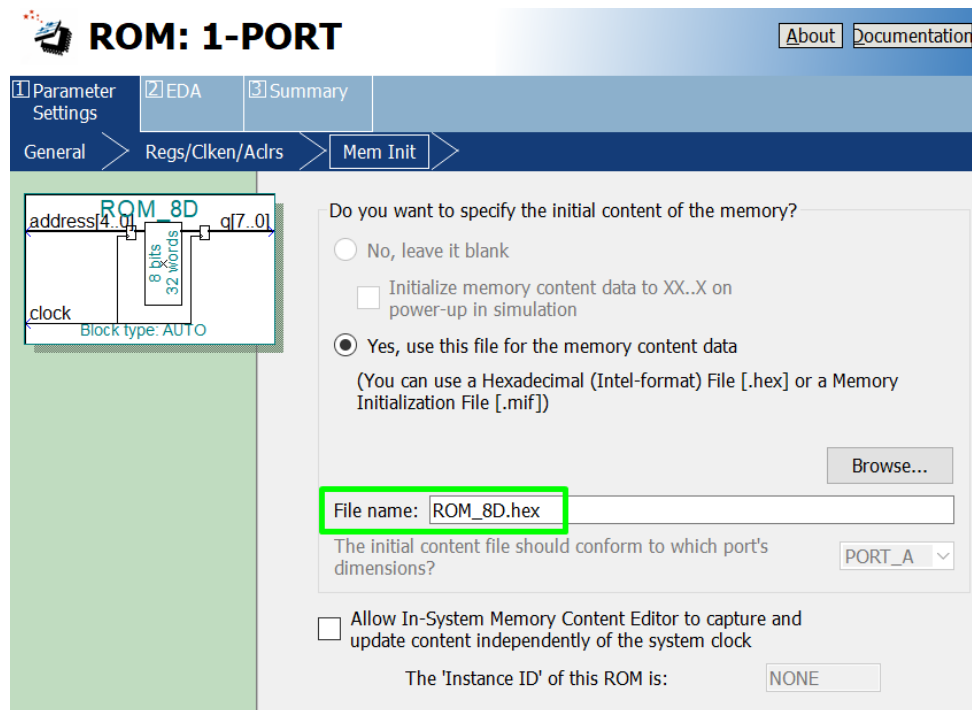


Figure 11

Щелкните **Next**.

g. В появившемся окне обратите внимание на указанную библиотеку **altera_mf**. Эта библиотека будет использована при моделировании созданного экземпляра IP модуля ROM-1Port. Щелкните **Next**.

h. В появившемся окне выберите **ROM_8D_inst.v** и **ROM_8D_bb.v**. Затем щелкните **Finish**.

i. В появившемся диалоговом окне щелкните **Yes** – к проекту будет добавлено описание созданного экземпляра IP модуля.

3 Чтобы проверить какие экземпляры IP модулей добавлены к проекту: в окне **Project navigator** выберите **IP Components**.

Project Navigator IP Components		
	Entity	IP Component
✓	cnt_adr	LPM_COUNTER
✓	ROM_8D	ROM: 1-PORT

Figure 12

Создайте остальные модули и файл верхнего уровня иерархии

- 1 Создайте описание модуля **DFF_chain** (имя файла: **DFF_chain.v**). Пример кода приведен на Figure 13.

```
1 module DFF_chain
2   (input  clk,
3    input  d,
4    output reg q );
5
6   reg d_int;
7   always @(posedge clk)
8   begin
9     d_int  <= d;
10    q      <= d_int;
11  end
12
13 endmodule
```

Figure 13

- 2 Создайте описание модуля **cnt_div** (имя файла: **cnt_div.v**) . Пример кода приведен на Figure 14.

```
1 module cnt_div
2   #(parameter div = 25)
3   (input clk,
4    input reset,
5    output reg cout );
6
7   reg [31:0] cnt;
8   wire cycle;
9
10  always @(posedge clk)
11    if (reset | cycle)
12      cnt <= 0;
13    else
14      cnt <= cnt+1;
15
16  assign cycle = (cnt == (div-1));
17
18  always @(posedge clk)
19    if (reset)
20      cout <= 1'b0;
21    else
22      if (cycle)
23        cout <= 1'b1;
24      else
25        cout <= 1'b0;
26
27 endmodule
```

Figure 14

- 3 Создайте описание модуля верхнего уровня иерархии **lab_QMS1** file (file: **lab_QMS1.v**).

Модуль lab_QMS1 – модуль верхнего уровня иерархии, объединяющий все ранее созданные модули в один проект, структура которого приведена на Figure 1. Пример кода приведен на Figure 15.


```

lab_QMS1.v X
C: > Intel_trn > Q_MS > lab_QMS1 > lab_QMS1.v
1  module lab_QMS1
2  #(parameter div_by = 25)
3  (input      CLK,
4  input      RESET,
5  output     [7:0] LED );
6
7  wire [4:0] adr;
8  wire clk_en;
9  wire srst;
10
11  DFF_chain DFF_chain_inst (
12  .clk      (CLK),
13  .d        (RESET),
14  .q        (srst)
15  );
16
17  cnt_div #(div_by) cnt_div_inst (
18  .clk      (CLK),
19  .reset    (RESET),
20  .cout     (clk_en)
21  );
22
23  cnt_adr cnt_adr_inst (
24  .clock    ( CLK),
25  .sclr    ( RESET),
26  .clk_en   ( clk_en),
27  .q        ( adr)
28  );
29
30  ROM_8D ROM_8D_inst (
31  .clock    ( CLK),
32  .address  ( adr),
33  .q        ( LED)
34  );
35
36  endmodule

```

Figure 15

- 4 Проверка проекта:
 - a. Если указанные выше файлы с описаниями модулей не были подключены к проекту – подключите их (Project => Add\Remove file in Project).
 - b. Выберите **Project navigator** => **Hierarchy**.
 - c. Выберите модуль **lab_QMS1**
 - d. Выполните **Processing** => **Start** => **Start Analysis and Elaboration**
 - e. По результатам компиляции не должно быть ошибок (Errors) и критических замечаний (Critical Warnings.)
- 5 Создайте описание теста **tb_lab_QMS1** (файл: **tb_lab_QMS1.v**).

```

C: > Intel_trn > Q_MS > lab_QMS1 > tb_lab_QMS1.v
1  `timescale 1ns/100ps
2  module tb_lab_QMS1();
3
4  reg tb_clk;
5  reg tb_reset;
6  wire [7:0] tb_led;
7
8  lab_QMS1 #(5)
9  DUT (
10     .LED      (tb_led      ),
11     .CLK      (tb_clk      ),
12     .RESET    (tb_reset    )
13 );
14 initial
15 begin
16     tb_clk      = 1'b0;
17     tb_reset    = 1'b1;
18
19     #100 tb_reset = 0;
20     #1000 $stop;
21 end
22
23 always #10 tb_clk = ~tb_clk;
24
25 endmodule

```

Figure 16

Настройка запуска ModelSim из пакета Quartus

- 1 Выполните команду **Tools** → **Options** → **EDA Tool Options**.
- 2 В разделе ModelSim-Altera должен отображаться путь к исполняемым файлам пакета ModelSim.

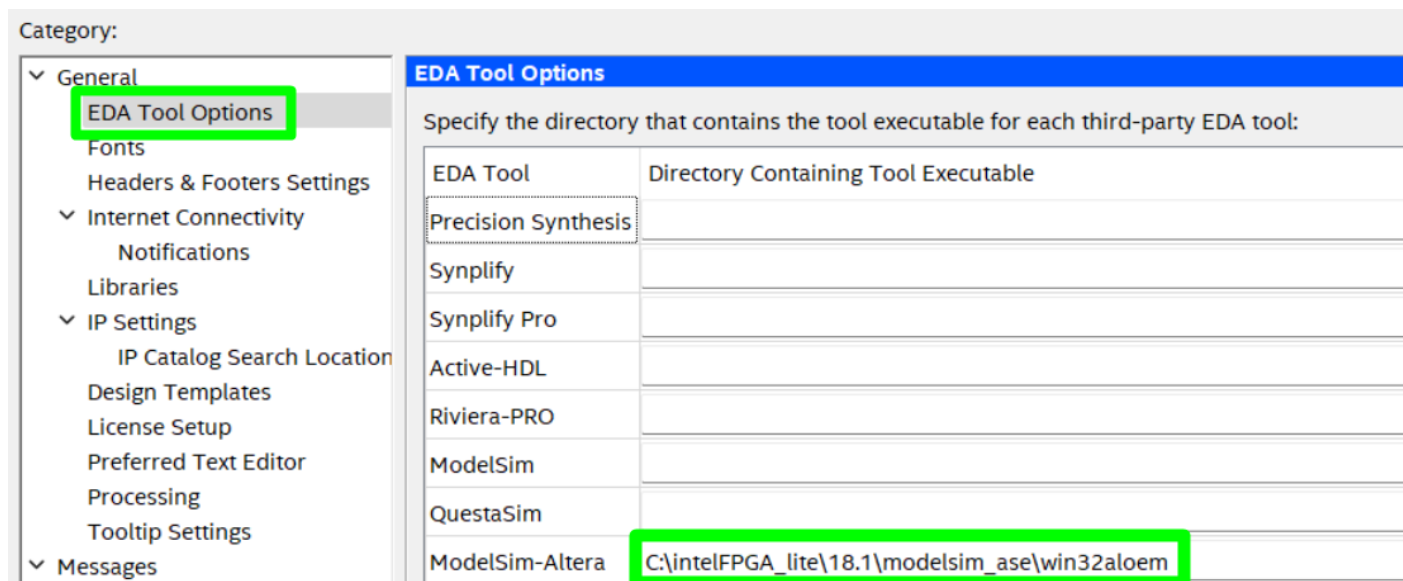
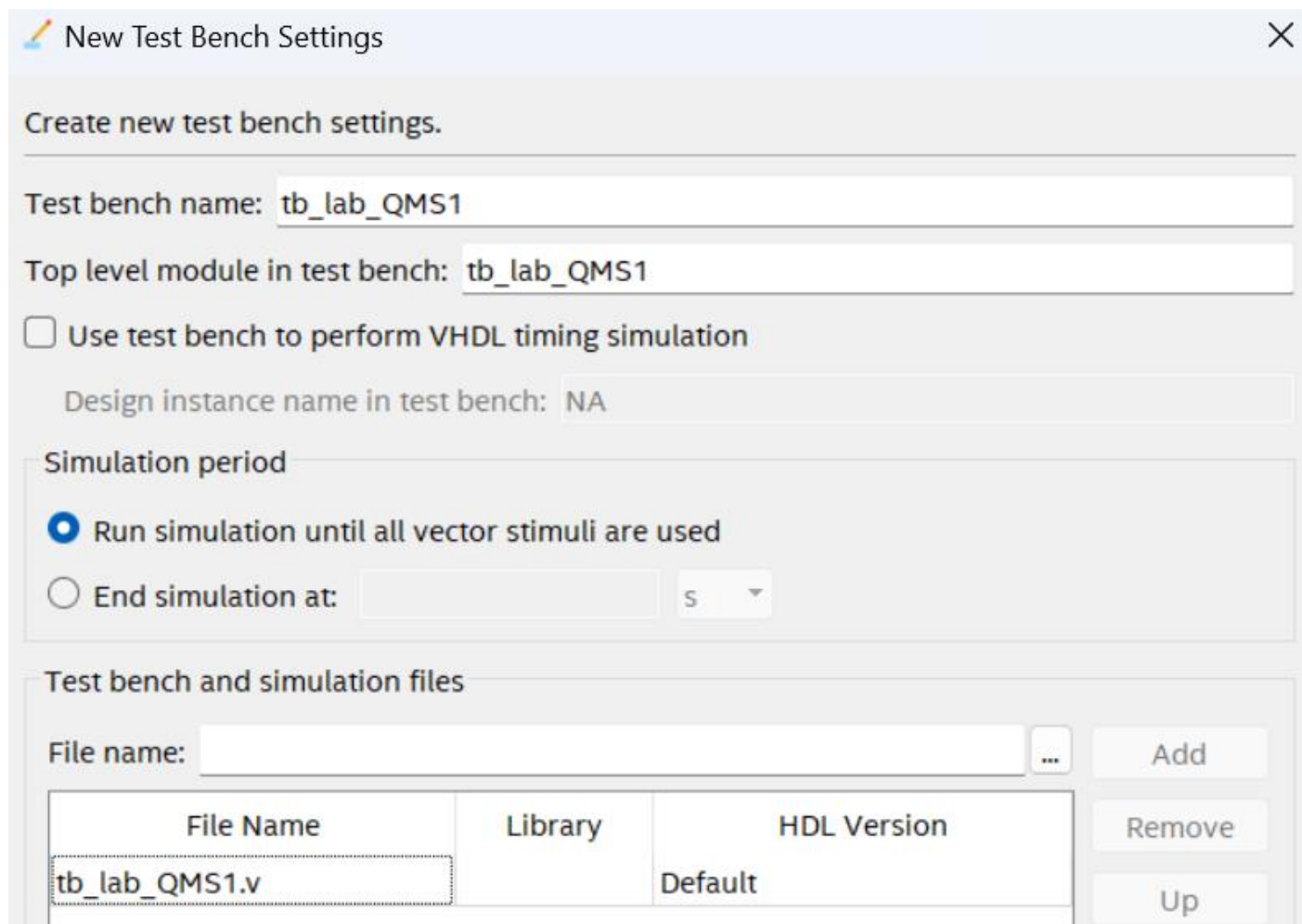


Figure 17

Щелкните **OK**.

- 3 Next step is to make sure that the test bench can be compiled without errors.
 - a. Выполните **Assignments** => **Settings** => **EDA Tool Settings** => **Simulation**.
 - b. Выберите **ModelSim-Altera** в поле **Tool Name**.
 - c. В разделе **NativeLink Settings** выберите **Compile Test Benches** и щелкните **Test Benches...**
 - d. В окне **Test Benches** щелкните **New...**
 - e. В поле Test bench name введите имя теста **tb_lab_QMS1**.
 - f. В поле **File name** выберите файл **tb_lab_QMS1.v**. Щелкните **Add**.



New Test Bench Settings

Create new test bench settings.

Test bench name:

Top level module in test bench:

☐ Use test bench to perform VHDL timing simulation

Design instance name in test bench:

Simulation period

☒ Run simulation until all vector stimuli are used

☐ End simulation at: s

Test bench and simulation files


File name: ...

File Name	Library	HDL Version
tb_lab_QMS1.v		Default

Add Remove Up

Figure 18

- g. Щелкните **OK**.
- h. Щелкните **OK** еще раз.
- i. Щелкните **OK** еще раз.

- 4 Вы полните команду **Start Analysis & Synthesis** .
- 5 Выполните команду **Tools=> Run Simulation=> RTL Simulation**
- 6 Пакет ModelSim будет запущен, будут установлены все библиотеки, в систему моделирования пакета будет загружен модуль **tb_lab_QMS1**, запущен тест. Окно Transcript и окно Wave будут отображать результаты моделирования.

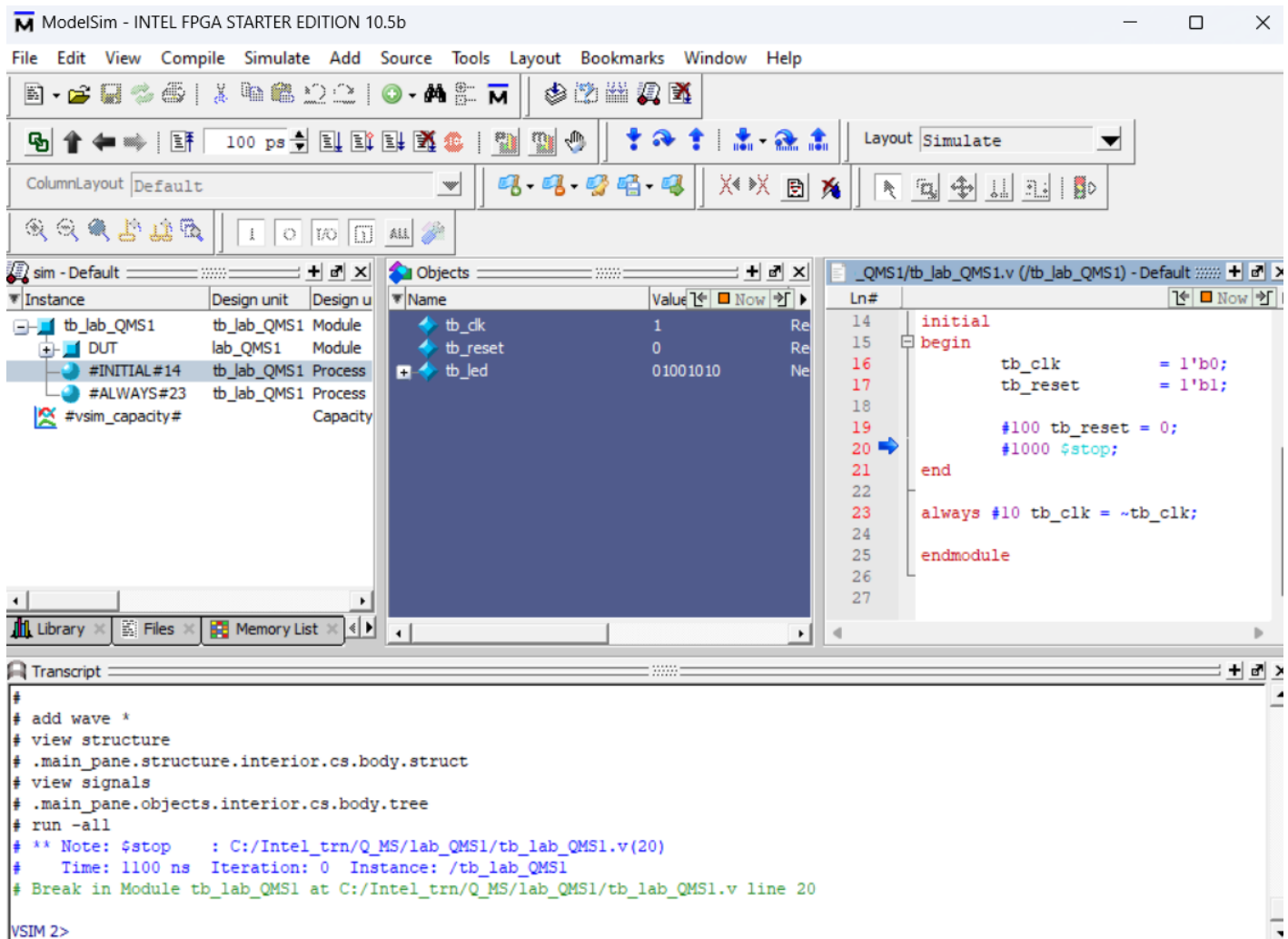




Figure 19

7 В окне Wave:

- Выполните Undo .
- Выполните Zoom Full .
- Для шины tb_led установите систему Radix как ASCII

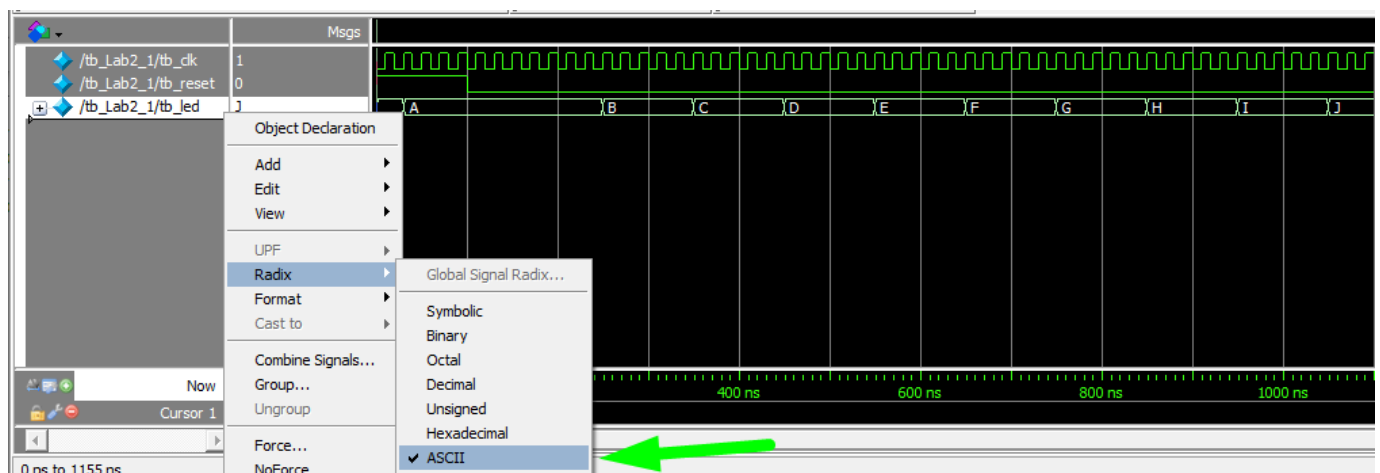


Figure 20

Ваша временная диаграмма должна быть похожа на диаграмму, приведенную на Figure 20.

Some tips:

Если нужно изменить масштаб временной оси (default timescale) например с ns на us, или наоборот: переместите курсор под временную ось и нажмите правую клавишу мыши. В проявившемся окне выберите **Grid, Timeline & Cursor Control..**

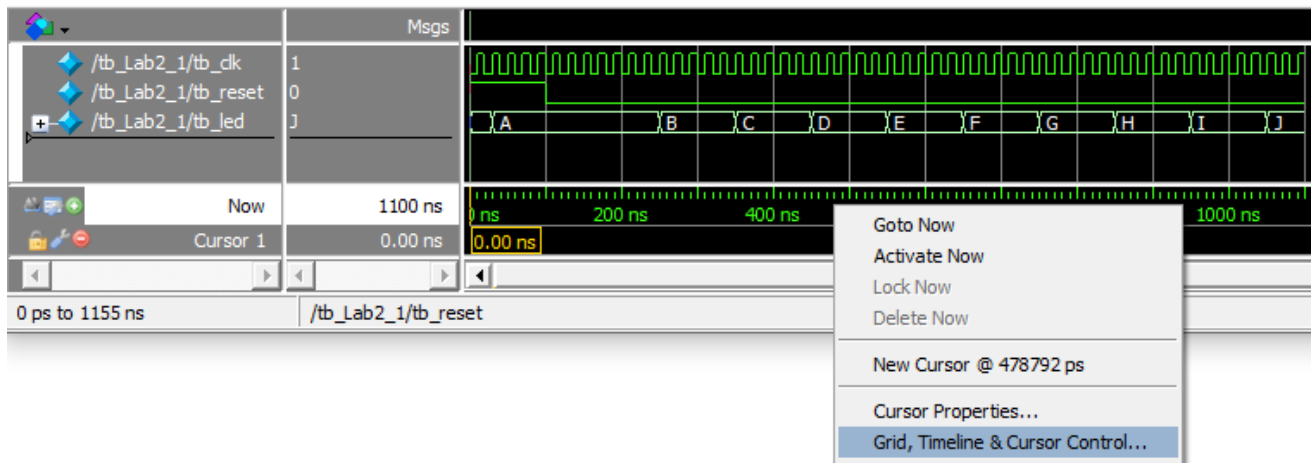


Figure 21

- В окне Grid, Timeline & Cursor Control, выберите us (или...) как Time Units (Figure 22).

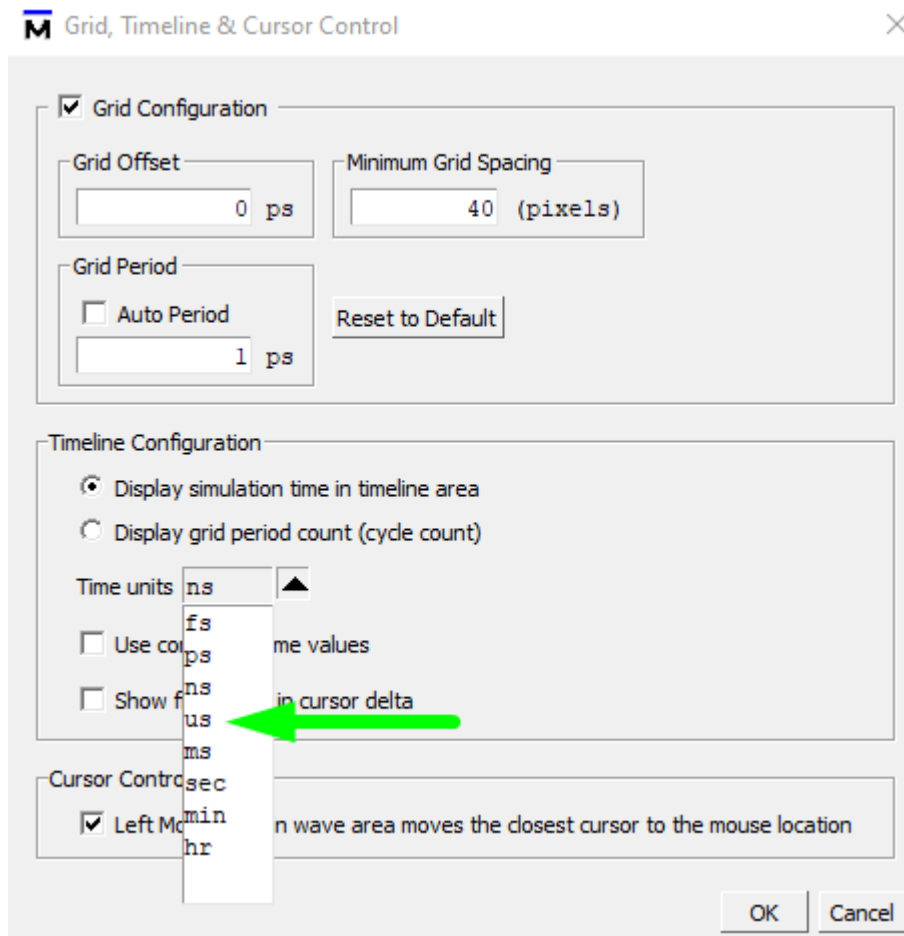


Figure 22

Щелкните **OK**.

Открытое окно пакета ModelSim можно использовать как пакет, запущенный обычным способом (независимо от пакета Quartus), до тех пор, пока нет необходимости изменить IP модули.

Т.е. можно менять любые (кроме IP модулей) исходные файлы, компилировать их, презапускать моделирование.... Если нужно изменить экземпляр IP модуля, то необходимо закрыть пакет ModelSim, перейти в пакет Quartus внести в экземпляр IP нужные изменения, осуществить компиляцию (Elaboration and Synthesis) и запустить из пакета Quartus пакет ModelSim: **Tools=> Run Simulation=> RTL Simulation**.

- Для сохранения заданных в окне Wave настроек выполните:
 - a. **File => Save Format.**
 - b. Задайте имя **wave_my_lab_QMS1.do**
 - c. Щелкните **Save**.
- 8 Завершите моделирование и закройте пакет ModelSim.

Дополнительные шаги

При первом запуске RTL Simulation из пакета Quartus в рабочей папке проекта создается папка **simulation**. В ней содержится папка **Modelsim** – рабочая папка для пакета ModelSim.

В папке **Modelsim** создается файл с расширением .do.

Его имя для нашего проекта будет **lab_QMS1_run_msim_rtl_verilog.do**.

Этот do файл можно запустить в пакете Modelsim независимо от пакета Quartus.

Для этого выполните следующие действия:

- Запустите пакет ModelSim независимо от пакета Quartus.
- В окне Transcript задайте рабочую папку :

```
cd C:/Intel_trn/Q_MS/lab_QMS1/simulation/modelsim
```

- В окне Transcript запустите файл do:

```
do lab_QMS1_run_msim_rtl_verilog.do
```

проект будет загружен без пакета Quartus.

- Закройте окно Wave
- В окне Transcript выполните команду

```
do wave_my_lab_QMS1.do
```

- Убедитесь в том, что полученная временная диаграмма похожа на диаграмму, приведенную на Figure 20.
- Завершите моделирование и закройте пакет ModelSim.
- Закройте пакет Quartus.

Часть 2 “Особенности использования IP из библиотеки пакета Quartus при моделировании в пакете ModelSim”

Запуск и настройка пакета ModelSim независимо от пакета Quartus

- 1 Запустите пакет ModelSim независимо от пакета Quartus.
- 2 Задайте рабочую папку.
 - a. Выполните **File => Change Directory** и укажите **C:\Intel_trn\Q_MS\lab_QMS1**.
*Или в окне Transcript выполните команду: **cd C:/Intel_trn/Q_MS/lab_QMS1***
- 3 Создайте рабочую библиотеку.
 - a. Выполните **File => New => Library => work**.
*Или в окне Transcript выполните команду: **vlib work***
- 4 Выполните компиляцию исходных файлов:
 - a. Выполните **Compile => Compile => "DFF_chain.v" "ROM_8D.v" "tb_lab_QMS1.v" "cnt_adr.v" "cnt_div.v" "lab_QMS1.v"**.
*Или в окне Transcript выполните команду: **vlog -work work "DFF_chain.v" "ROM_8D.v" "tb_lab_QMS1.v" "cnt_adr.v" "cnt_div.v" "lab_QMS1.v"***
- 5 Убедитесь в отсутствии ошибок. Если появятся ошибки – надо внести исправления.
- 6 Загрузите модуль теста в систему моделирования:
 - a. В окне Library щелкните знак ‘+’ около библиотеки **work** и дважды щелкните **tb_lab_QMS1**.
*Или в окне Transcript выполните команду: **vsim work.tb_lab_QMS1***
- 7 В окне Transcript появится сообщение об ошибке

```
ModelSim> vsim work.tb_lab_QMS1
# vsim work.tb_lab_QMS1
# Start time: 21:28:58 on Nov 10, 2024
# Loading work.tb_lab_QMS1
# Loading work.lab_QMS1
# Loading work.DFF_chain
# Loading work.cnt_div
# Loading work.cnt_adr
# ** Error: (vsim-3033) cnt_adr.v(53): Instantiation of 'lpm_counter' failed. The design unit was not found.
#   Time: 0 ps Iteration: 0 Instance: /tb_lab_QMS1/DUT/cnt_adr_inst File: cnt_adr.v
#   Searched libraries:
#     C:/Intel_trn/Q_MS/lab_QMS1/work
# Loading work.ROM_8D
# ** Error: (vsim-3033) ROM_8D.v(58): Instantiation of 'altsyncram' failed. The design unit was not found.
#   Time: 0 ps Iteration: 0 Instance: /tb_lab_QMS1/DUT/ROM_8D_inst File: ROM_8D.v
#   Searched libraries:
#     C:/Intel_trn/Q_MS/lab_QMS1/work
# Error loading design
# End time: 21:28:58 on Nov 10, 2024, Elapsed time: 0:00:00
# Errors: 2, Warnings: 0
```

Figure 23

- 8 Причина ошибки в том, что не были указаны библиотеки **lpm** и **altera_mf**. Укажите их:
 - a. Выберите **Simulate => Start Simulation**
 - b. В окне щелкните знак ‘+’ около библиотеки **work** и выберите **tb_lab_QMS1**.
 - c. На закладке **Libraries**
 - i В поле Search Libraries щелкните **Add...**
 - ii В появившемся окне выберите библиотеку **altera_mf_ver**

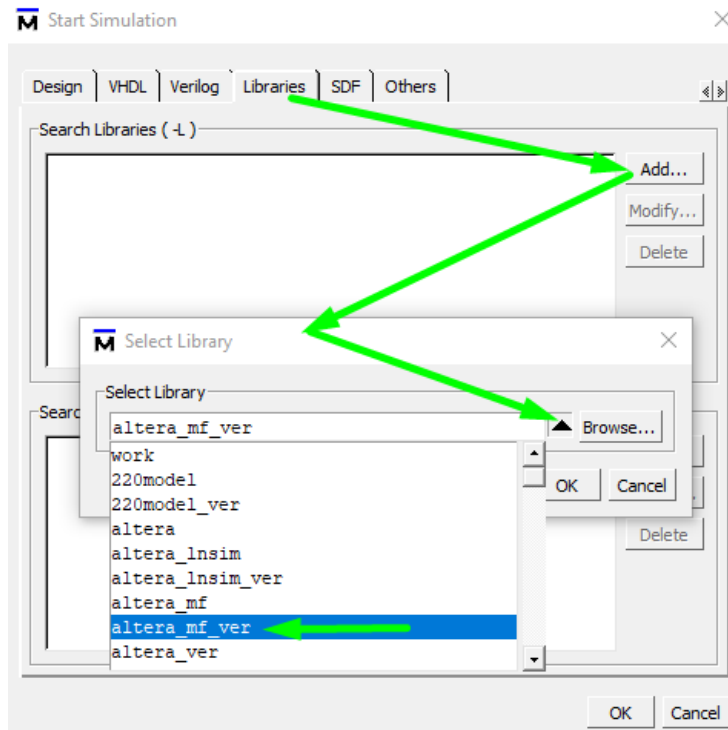


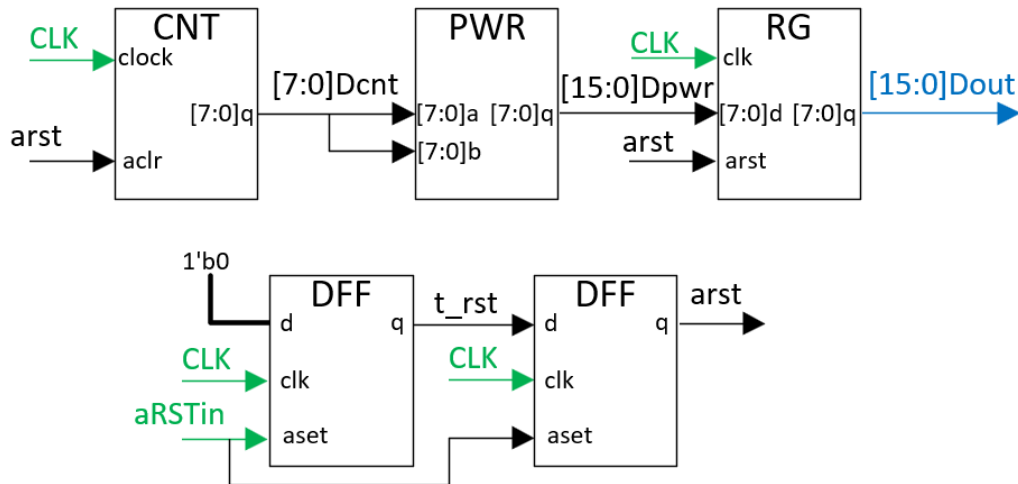
Figure 24

- iii Щелкните OK
 - iv Щелкните **Add...** еще раз
 - v Выберите библиотеку **lpm_ver**.
 - vi Щелкните OK
 - vii Щелкните OK еще раз
 - viii Модуль теста должен быть загружен без ошибок.
- d. В окне Transcript введите команду:
- do C:/Intel_trn/Q_MS/lab_QMS1/simulation/modelsim/wave_my_lab_QMS1.do**
- Откроется окно Wave с настроенным ранее форматом.
- e. В окне transcript выполните команду: **run 1000 ns**
- f. Ваша временная диаграмма должна быть похожа на диаграмму, приведенную на Figure 20
- 9 Завершите моделирование и закройте пакет ModelSim.

Lab_QMS2

(использование ModelSim независимо от Quartus, как было описано в части 2 работы lab_QMS1)

Структура проекта



1. Выводы

- a. Входы (отмечены зеленым)
 - i. CLK – тактовый сигнал (25МГц)
 - ii. aRSTin – вход асинхронного сброса (активный уровень, уровень при котором будет сброс – 1)
- b. Выходы (отмечены синим)
 - i. [15:0] Dout – выход.

2. Модули

- a. **CNT** - счетчик, создаваемый с помощью IP модуля **LPM_COUNTER**
 - i. Разрядность: 8 бит
 - ii. Двоичный счетчик на сложение
 - iii. Вход асинхронного сброса (clear) (активный уровень 1)
- b. **PWR** – модуль возведения в степень 2, создаваемый с помощью IP модуля **LPM_MULT**.
 - i. Два входа по 8 бит
 - ii. Без знаковый
 - iii. Без конвейеризации
- c. **RG** – регистр, описываемый на Verilog в файле верхнего уровня (используя always)
 - i. arst – вход асинхронного сброса (активный уровень -1).
- d. **DFF** – триггеры, описываемые на Verilog в файле верхнего уровня (используя always)
 - i. Логическая единица на входе aset – асинхронно устанавливает триггер в 1.

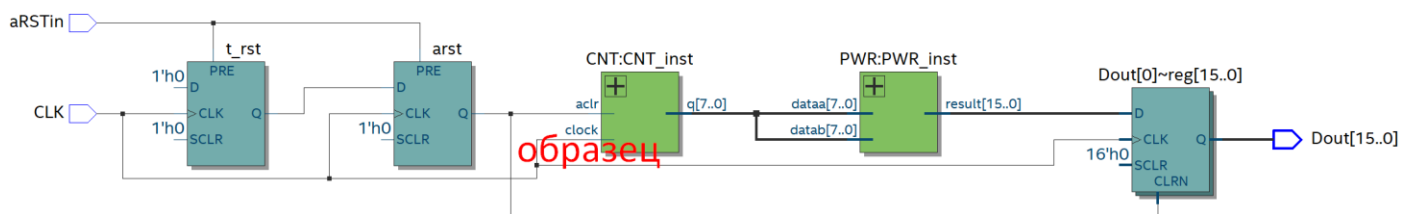
План работы

1. Создать проект в пакете Quartus
 - a. Рабочая папка C:\Intel_trn\Q_MS\lab_QMS2

- b. Имя проекта **lab_QMS2**
 - c. Модуль верхнего уровня **lab_QMS2**
 - d. Микросхема **EP4CE6E22C8**
 - e. **Внешние средства проектирования не задавать.**
2. Создать модули на основе IP

СОВЕТ: при создании модулей надо запомнить библиотеки, которые должны быть подключены для моделирования.

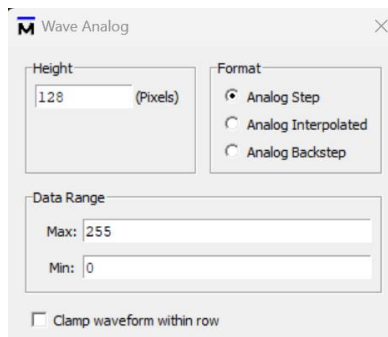
3. Создать модуль верхнего уровня иерархии на Verilog (имя файла lab_QMS2.v, модуль lab_QMS2).
4. Осуществить компиляцию проекта в пакете Quartus, исправить ошибки и проверить с помощью RTL Viewer, что проект собран правильно (соответствует структуре в задании и образцу ниже).



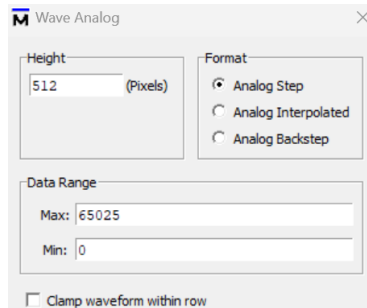
5. Разработать тест класса 1, обеспечивающий проверку для всех значений счетчика CNT (– имя файла tb_lab_QMS2.v, имя модуля – tb_lab_QMS2
 - a. надо запускать тест как минимум на 2 полных цикла счета счетчика.
6. Запустить пакет ModelSim отдельно от пакета Quartus
7. При желании работать с проектом: создать проект в пакете ModelSim
 - a. Включить исходные файлы в проект
8. Осуществить компиляцию исходных файлов, включая тест.
 - a. Если появятся ошибки их надо исправить.
9. Загрузить тест tb_lab_QMS2 в систему моделирования (либо, если работаете с проектом: создать конфигурацию для моделирования и загрузить проект в систему моделирования)

СОВЕТ: не забудьте о том, что надо подключить библиотеки для моделирования.

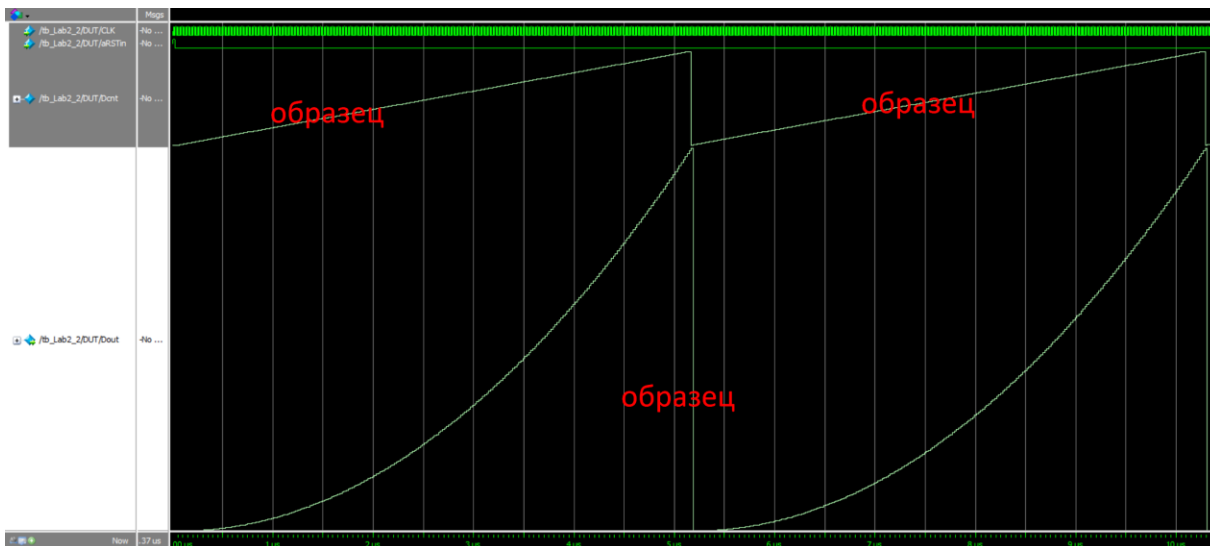
10. Осуществить моделирование
 - a. Проверить правильность работы устройства, при необходимости отладить его.
 - b. На временную диаграмму надо вывести сигналы CLK, aRSTin, Dcnt, Dout, представленные на структуре.
 - c. Для шин Dcnt и Dout следует задать RADIX: unsigned
 - d. Для шины Dcnt надо задать Format: analog(custom), например, с параметрами, указанными ниже.



- е. Для шины Dout надо задать Format: analog(custom), например, с параметрами, указанными ниже.



- ф. Временная диаграмма должна быть похожа на образец ниже.



11. Сохранить до файл с настройками окна временных диаграмм (при показе преподавателю – загрузить до файл).

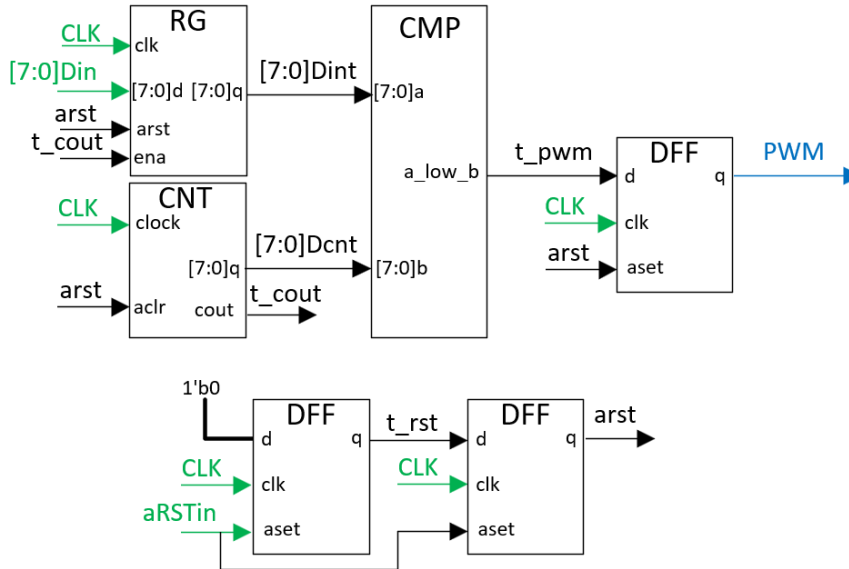
Что должен включать отчет.

1. Задание.
2. Исходный код файла lab_ЙБЫ2.v. (с пояснением)
3. Структуру из RTL viewer (с анализом соответствия исходной структуре задания).
4. Исходный код теста (с пояснениями).
5. Результаты моделирования (с анализом результатов, временной диаграммой и пояснениями).
6. Выводы.

Lab_QMS3

(использование NativeLink пакета Quartus для запуска ModelSim, как было описано в части 1 работы lab_QMS1)

Структура проекта



1. Выводы

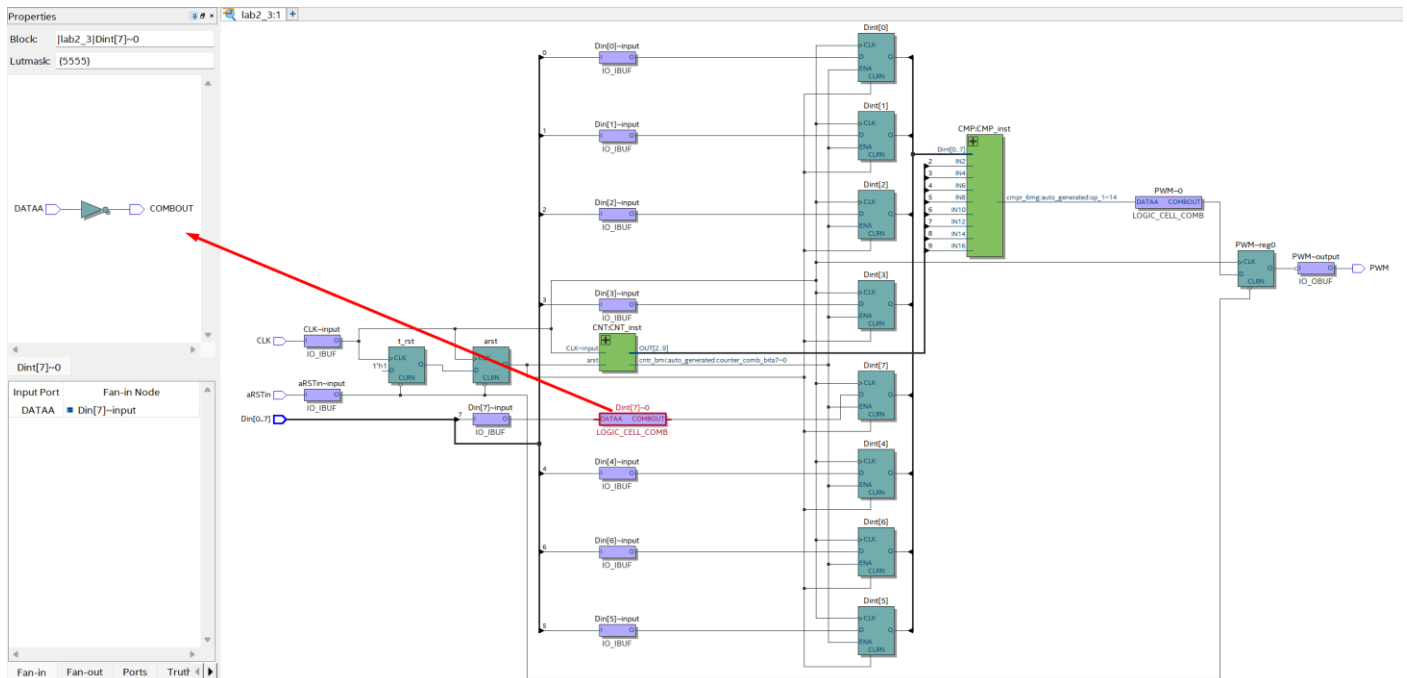
- a. Входы (отмечены зеленым)
 - i. CLK – тактовый сигнал (50МГц).
 - ii. aRSTin – вход асинхронного сброса (активный уровень, уровень при котором будет сброс – 1).
 - iii. [7:0]Din – вход данных для управления ШИМ.
- b. Выходы (отмечены синим)
 - i. PWM – выход ШИМ.

2. Модули

- a. **CNT** - счетчик, создаваемый с помощью IP модуля **LPM_COUNTER**
 - i. Разрядность: 8 бит
 - ii. Двоичный счетчик на сложение
 - iii. Выход переноса (carry_out)
 - iv. Вход асинхронного сброса (clear) (активный уровень 1)
- b. **CMP** – модуль сравнения, создаваемый с помощью IP модуля **LPM_COMPARE**.
 - i. Два входа по 8 бит
 - ii. $a < b$
 - iii. Без знаковый
 - iv. Без конвейеризации
- c. **RG** – регистр, описываемый на Verilog в файле верхнего уровня (используя always)
 - i. arst – вход (активный уровень -1) – асинхронно устанавливает в регистр значение 8'd128.
- d. **DFF** – триггеры, описываемые на Verilog в файле верхнего уровня (используя always)
 - i. Логическая единица на входе aset – асинхронно устанавливает триггер в 1.

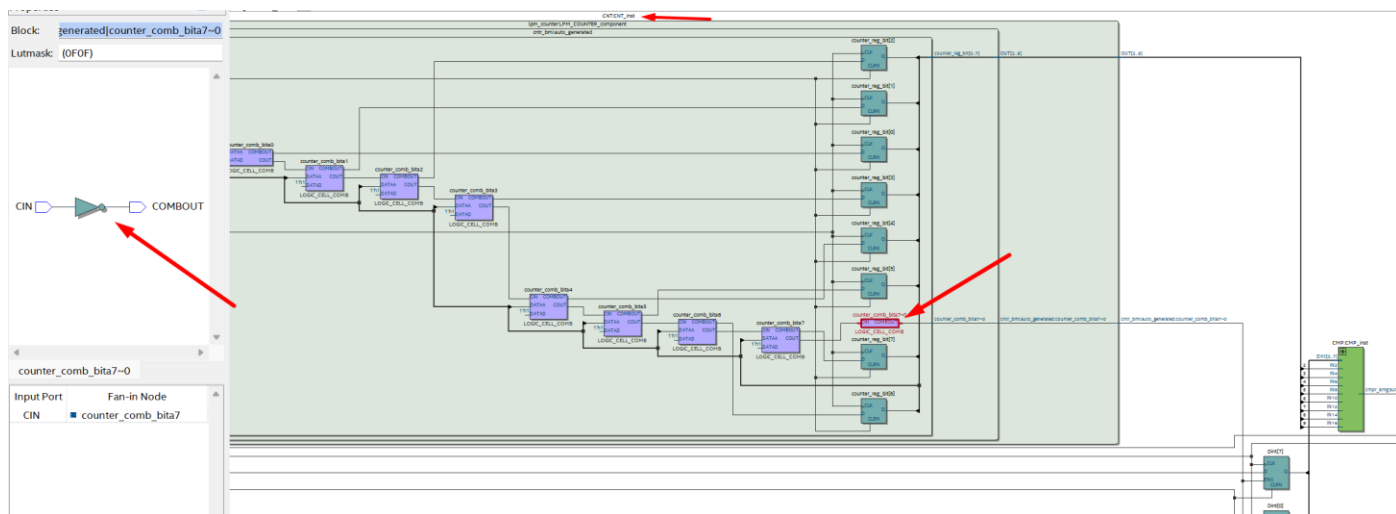
План работы

1. Создать проект в пакете Quartus
 - a. Рабочая папка C:\Intel_trn\Q_MS\lab_QMS3
 - b. Имя проекта lab_QMS3
 - c. Модуль верхнего уровня lab_QMS3
 - d. Микросхема EP4CE6E22C8
 - e. Внешнее средство проектирования – ModelSim-Altera Edition.
 - i. Язык - Verilog HDL
2. Создать модули на основе IP
3. Создать модуль верхнего уровня иерархии на Verilog (имя файла lab_QMS3.v, модуль lab_QMS3).
4. Осуществить компиляцию проекта в пакете Quartus, исправить ошибки и проверить с помощью Technology Map Viewer, что проект собран правильно (соответствует структуре в задании и образцу ниже).
 - a. **ОБЪЯСНИТЕ:** почему на входе разряда [7] регистра Din появился инвертор?



ПОДСКАЗКА: для этого надо

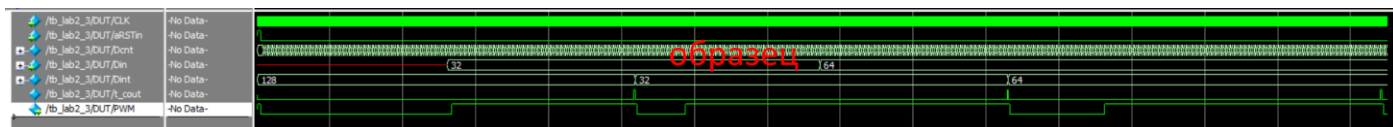
- вспомнить: что записывается в разряд [7] в исходном коде?
- посмотреть реализацию сброса/установки разряда [7] в Technology Map Viewer
- раскрыть в Technology Map Viewer модуль CNT и найти указанный на картинке ниже инвертор



5. Разработать тест класса 1 (имя файла `tb_lab_QMS3.v`, имя модуля – `tb_lab_QMS3`)
 - a. надо запускать тест на 3 полных цикла счета счетчика.
 - b. Задать разные значения `Din` для каждого цикла счета счетчика (см. образец временной диаграммы ниже)
6. Запустить пакет ModelSim используя NativeLink пакета Quartus
 - a. Если появятся ошибки их надо исправить.
7. Осуществить моделирование
 - a. Проверить правильность работы устройства, при необходимости отладить его.
 - b. На временную диаграмму надо вывести сигналы `CLK`, `aRSTin`, `Dcnt`, `Din`, `Dint`, `t_cout`, `PWM` представленные на структуре в задании.
 - c. Для всех шин следует задать RADIX: unsigned
 - d. Полученная временная диаграмма должна быть похожа на образец ниже.

ОБЪЯСНИТЕ:

- Как зависит выход PWM от данных `Dint`? Что за устройство реализовано?
- В какой момент считываются входные данные (данные со входа `Din`)?



8. Сохраните до файл с настройками окна временных диаграмм (при показе преподавателю – загрузить до файл).

Что должен включать отчет.

1. Задание.
2. Исходный код файла `lab_QMS3.v`. (с пояснением)
3. Структуру из RTL viewer (с анализом соответствия исходной структуре задания).
4. Исходный код теста (с пояснениями).
5. Результаты моделирования (с анализом результатов, временной диаграммой и пояснениями).
6. Выводы + ОТВЕТЫ НА ВОПРОСЫ, ПРИВЕДЕННЫЕ В ТЕКСТЕ ВЫШЕ.