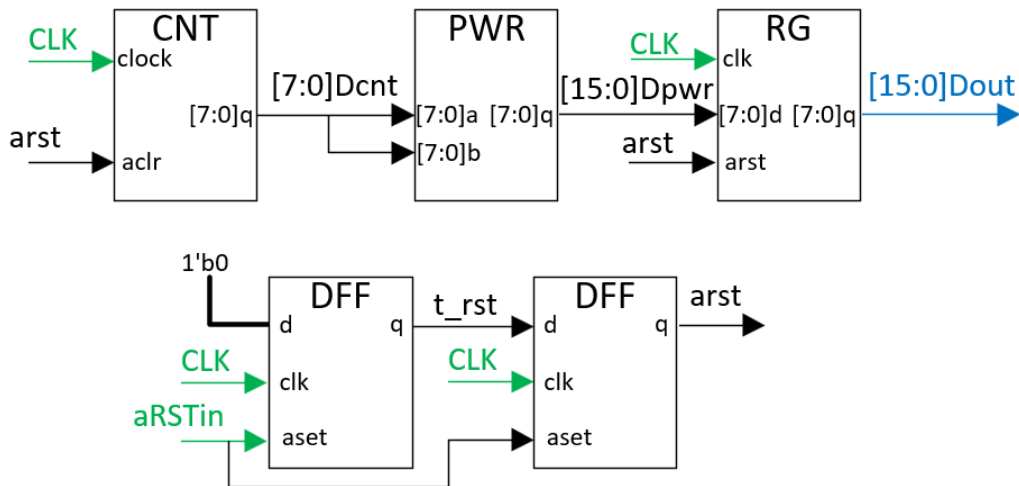


Lab2_2

(использование ModelSim независимо от Quartus, как было описано в части 2 работы lab2_1)

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)
- b. Выходы (отмечены синим)
 - i. [15:0] Dout – выход.

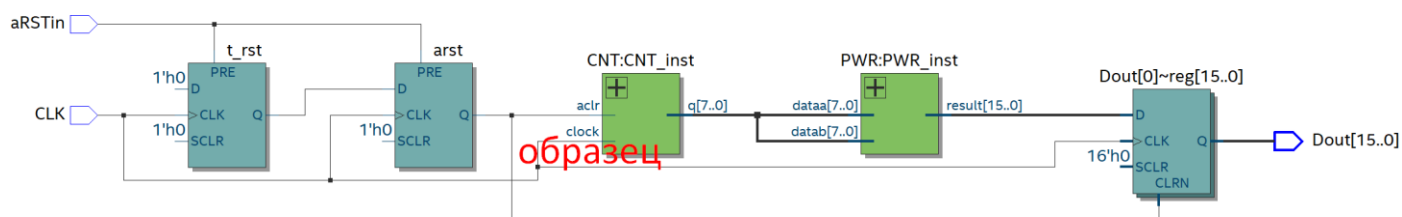
2. Модули

- 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.

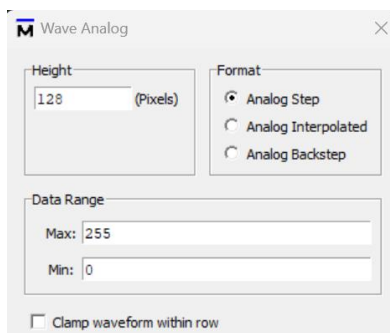
План работы

1. Создать проект в пакете Quartus
 - a. Рабочая папка **C:\Intel_trn\Q_MS\Lab2_2**
 - b. Имя проекта **lab2_2**
 - c. Модуль верхнего уровня **lab2_2**
 - d. Микросхема **EP4CE6E22C8**
 - e. **Внешние средства проектирования не задавать.**
2. Создать модули на основе IP
 - a. **СОВЕТ:** при создании модулей надо запомнить библиотеки, которые должны быть подключены для моделирования.
3. Создать модуль верхнего уровня иерархии на Verilog (имя файла Lab2_2.v, модуль lab2_2).

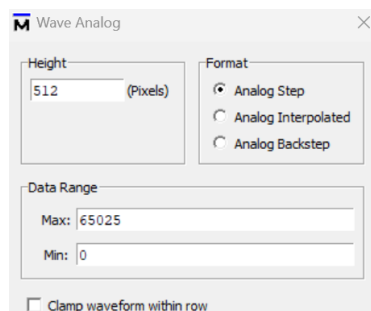
4. Осуществить компиляцию проекта в пакете Quartus, исправить ошибки и проверить с помощью RTL Viewer, что проект собран правильно (соответствует структуре в задании и образцу ниже).



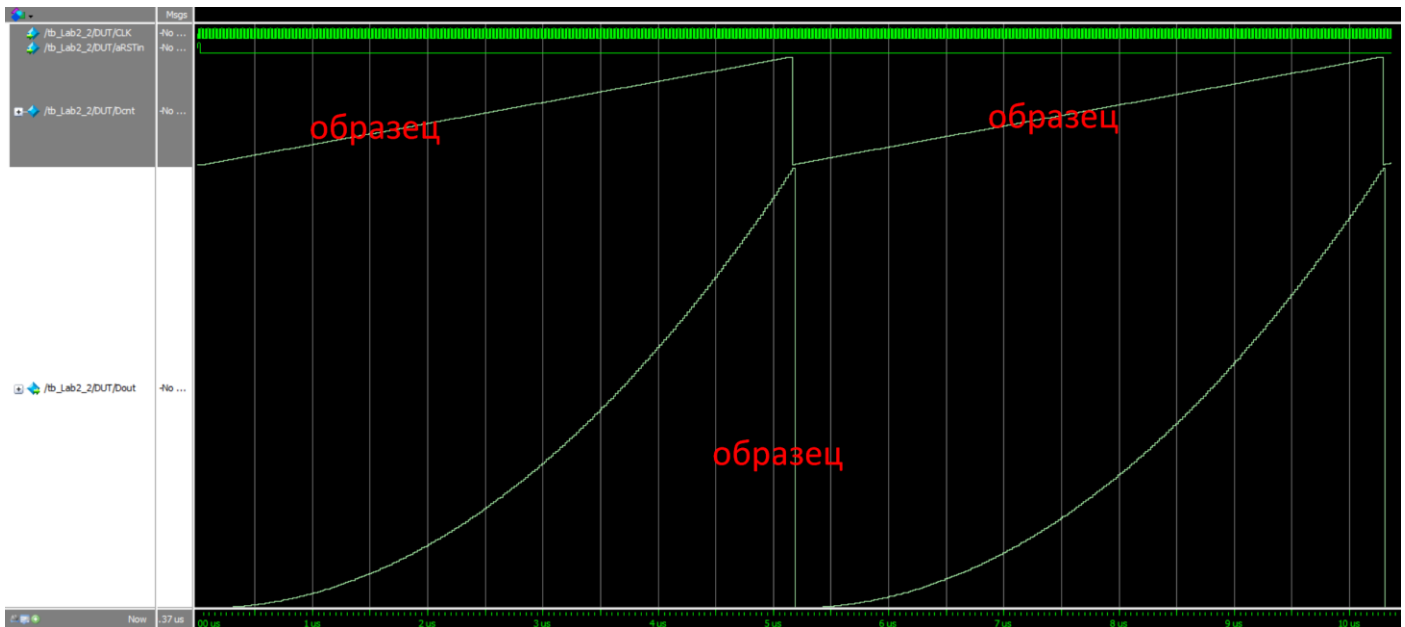
5. Разработать тест класса 1, обеспечивающий проверку для всех значений счетчика CNT (– имя файла tb_lab2_2.v, имя модуля – tb_lab2_2
 - а. надо запускать тест как минимум на 2 полных цикла счета счетчика.
6. Запустить пакет ModelSim отдельно от пакета Quartus
7. При желании работать с проектом: создать проект в пакете ModelSim
 - а. Включить исходные файлы в проект
8. Осуществить компиляцию исходных файлов, включая тест.
 - а. Если появятся ошибки их надо исправить.
9. Загрузить тест tb_lab2_2 в систему моделирования (либо, если работаете с проектом: создать конфигурацию для моделирования и загрузить проект в систему моделирования)
 - а. СОВЕТ: не забудьте о том, что надо подключить библиотеки для моделирования.
10. Осуществить моделирование
 - а. Проверить правильность работы устройства, при необходимости отладить его.
 - б. На временную диаграмму надо вывести сигналы CLK, aRSTin, Dcnt, Dout, представленные на структуре.
 - в. Для шин Dcnt и Dout следует задать RADIX: unsigned
 - д. 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.



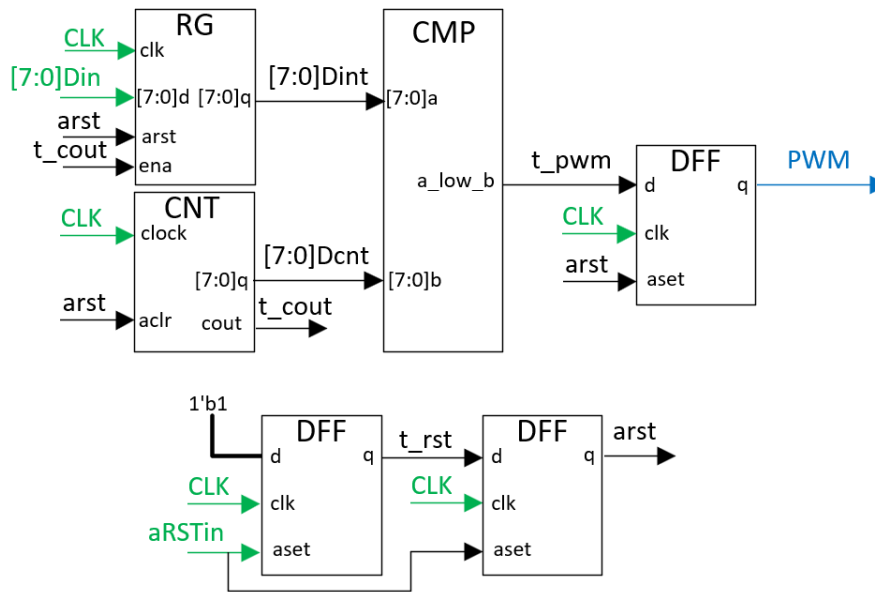
11. Сохранить do файл с настройками окна временных диаграмм (при показе преподавателю – загрузить do файл).

Что должен включать отчет.

1. Задание.
2. Исходный код файла `lab2_2.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.

(использование NativeLink пакета Quartus для запуска ModelSim, как было описано в части 1 работы lab2_1)

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. **CMР** – 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.

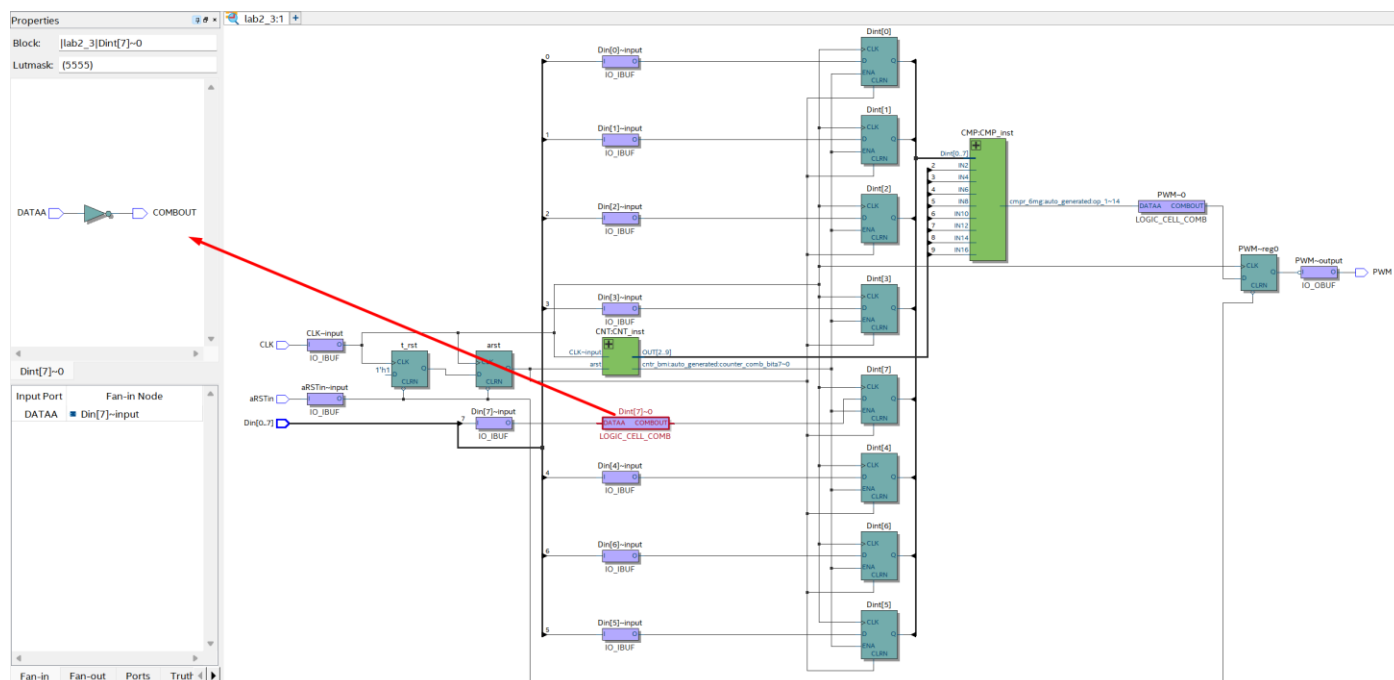
План работы

1. Создать проект в пакете Quartus
 - a. Рабочая папка **C:\Intel_trn\Q_MS\Lab2_3**
 - b. Имя проекта **lab2_3**
 - c. Модуль верхнего уровня **lab2_3**
 - d. Микросхема **EP4CE6E22C8**

f. **Язык - Verilog HDL**

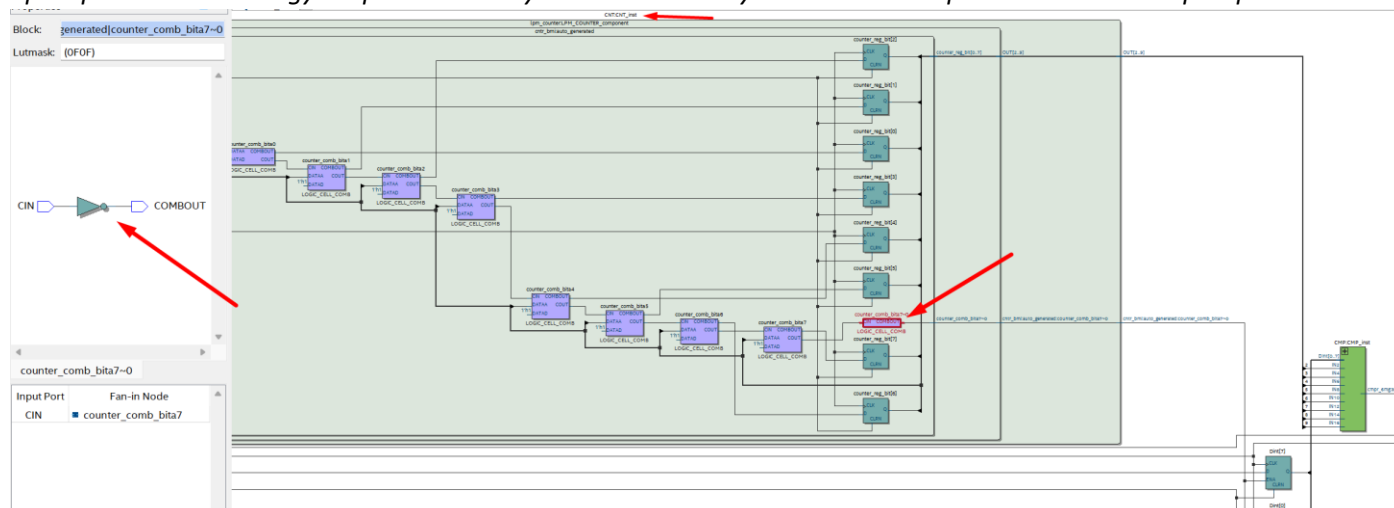
2. Создать модули на основе IP
3. Создать модуль верхнего уровня иерархии на Verilog (имя файла Lab2_3.v, модуль lab2_3).
4. Осуществить компиляцию проекта в пакете Quartus, исправить ошибки и проверить с помощью Technology Map Viewer, что проект собран правильно (соответствует структуре в задании и образцу ниже).

а. **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



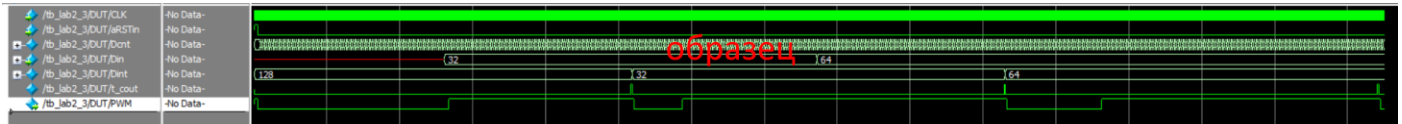
5. Разработать тест класса 1 (имя файла tb_lab2_3.v, имя модуля – tb_lab2_3
 - а. надо запускать тест на 3 полных цикла счета счетчика.
 - б. Задать разные значения Din для каждого цикла счета счетчика (см. образец временной диаграммы ниже)
6. Запустить пакет ModelSim используя NativeLink пакета Quartus
 - а. Если появятся ошибки их надо исправить.

7. Осуществить моделирование

- Проверить правильность работы устройства, при необходимости отладить его.
- На временную диаграмму надо вывести сигналы CLK, aRSTin, Dcnt, Din, Dint, t_cout, PWM представленные на структуре в задании.
- Для всех шин следует задать RADIX: unsigned
- 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 do файл с настройками окна временных диаграмм (при показе преподавателю – загрузить do файл).

Что должен включать отчет.

- Задание.
- Исходный код файла lab2_3.v. (с пояснением)
- Структуру из RTL viewer (с анализом соответствия исходной структуре задания).
- Исходный код теста (с пояснениями).
- Результаты моделирования (с анализом результатов, временной диаграммой и пояснениями).
- Выводы + ОТВЕТЫ НА ВОПРОСЫ, ПРИВЕДЕННЫЕ В ТЕКСТЕ ВЫШЕ.