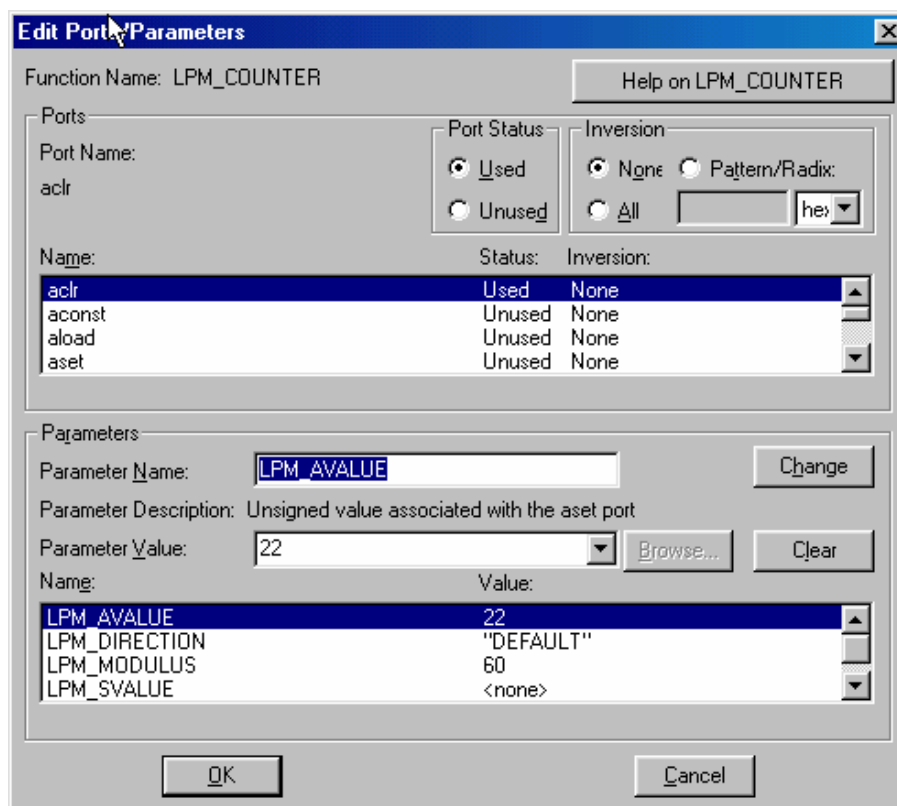


Входные и выходные порты и параметры счетчика.



Ниже приводится полный список всех возможных входных и выходных портов. Всего 17 позиций. В этом списке нужно последовательно выделить и пометить порты, которые необходимы для разрабатываемого проекта.

Входные порты.

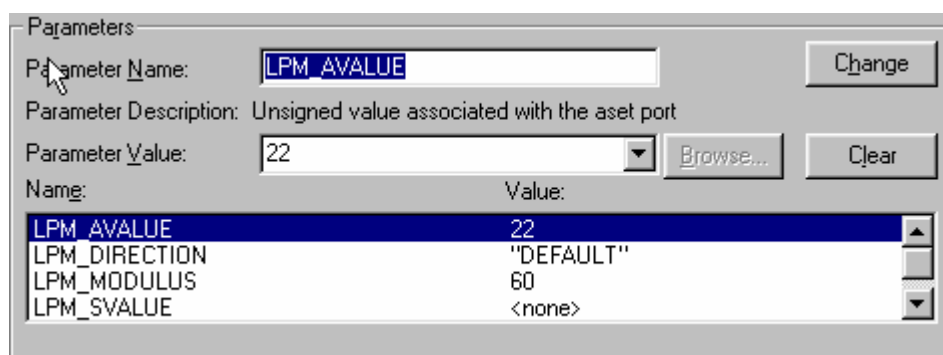
1. **aclr** – асинхронный сброс в «0» выходного кода.
2. **asset** – асинхронная установка одного из двух значений выходного кода. Если параметр **LPM_AVALUE** задан, то выходной код соответствует значению этого параметра. Если параметр **LPM_AVALUE** не задан, то во всех разрядах выходного кода устанавливается «1».
3. **aload** – асинхронная установка на выходе счётчика кода, действующего на входе предварительной установки **data[]**
4. **clk_en** – разрешение активизации тактирующих импульсов **clock**.
5. **clock** – вход тактирующих импульсов.
6. **sclr** – синхронная установка «0» во всех разрядах выходного кода.
7. **sset** – синхронная установка одного из двух значений выходного кода. . Если параметр **LPM_SVALUE** задан, то выходной код соответствует значению этого параметра. Если этот параметр не задан, то во всех разрядах устанавливается «1». В отличие от порта **asset** здесь изменения кода «привязаны» к тактирующим импульсам **clock**.
8. **sload** - асинхронная установка на выходе счётчика кода, действующего на входе предварительной установки **data[]**
9. **cnt_en** – разрешение счёта.
10. **updown** – вход управления направлением счёта
11. **data[]** – входной порт начальной установки счётчика. Используется совместно с **aload** или с **sload**.
12. **cin** – вход переноса (**Carry-in to the low-order bit**). Если не задействован, то по умолчанию «0»
13. **aconst** – В **Help on LPM_COUNTER** комментарии отсутствуют.
14. **sconst** - В **Help on LPM_COUNTER** комментарии отсутствуют

Выходные порты

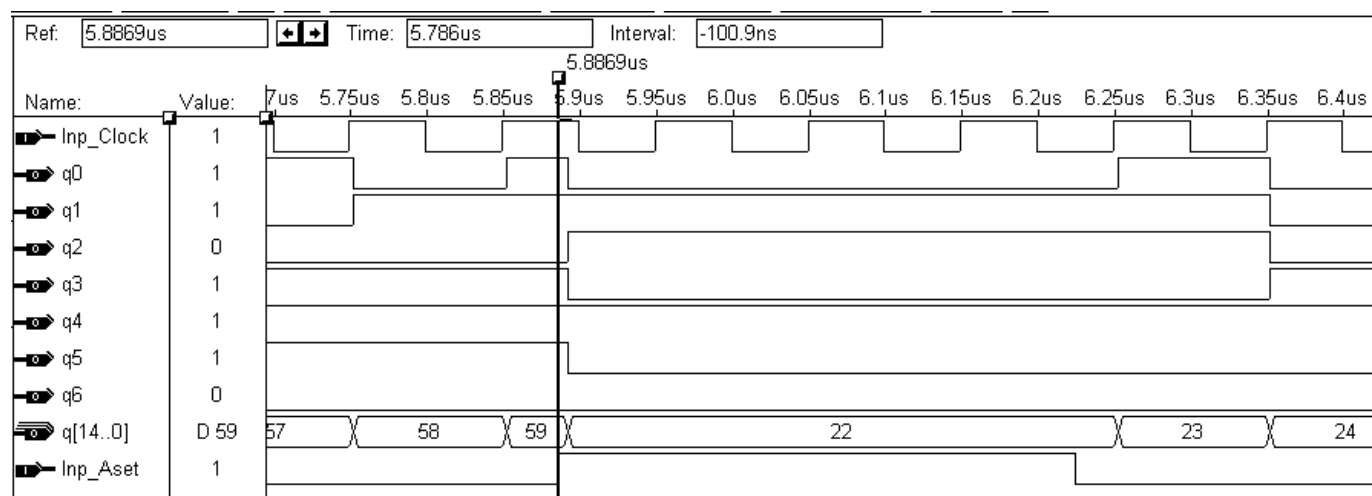
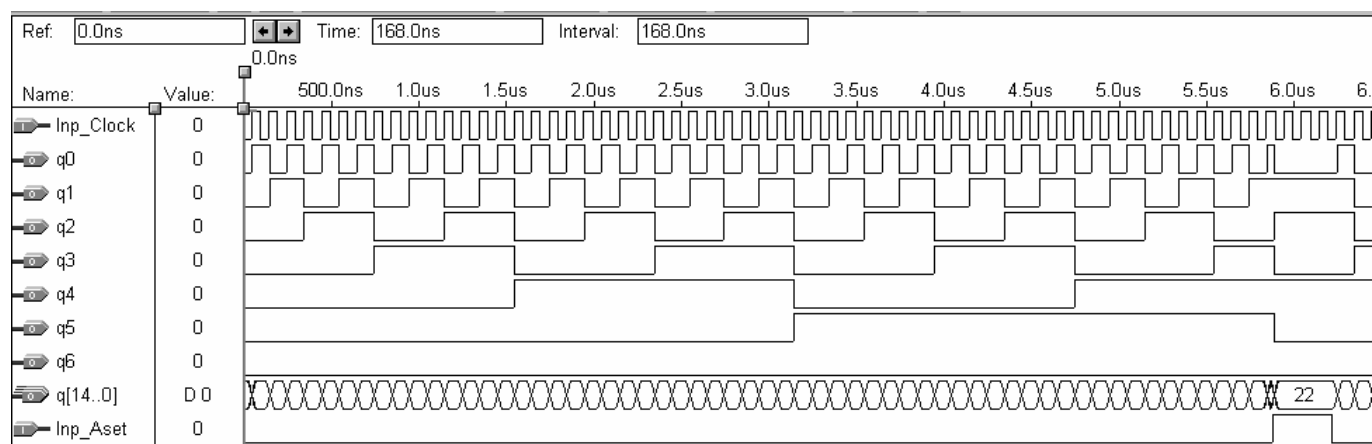
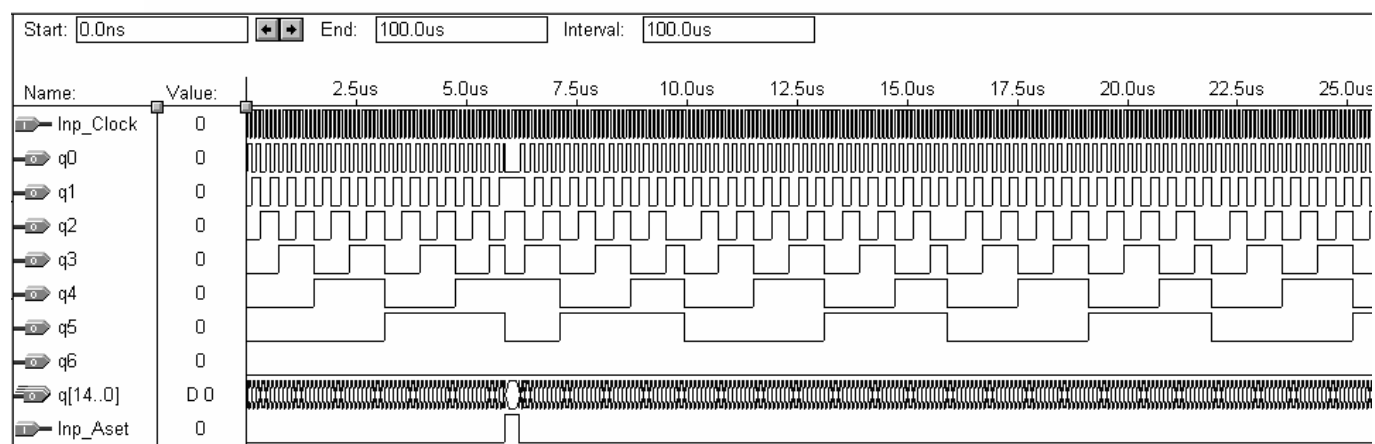
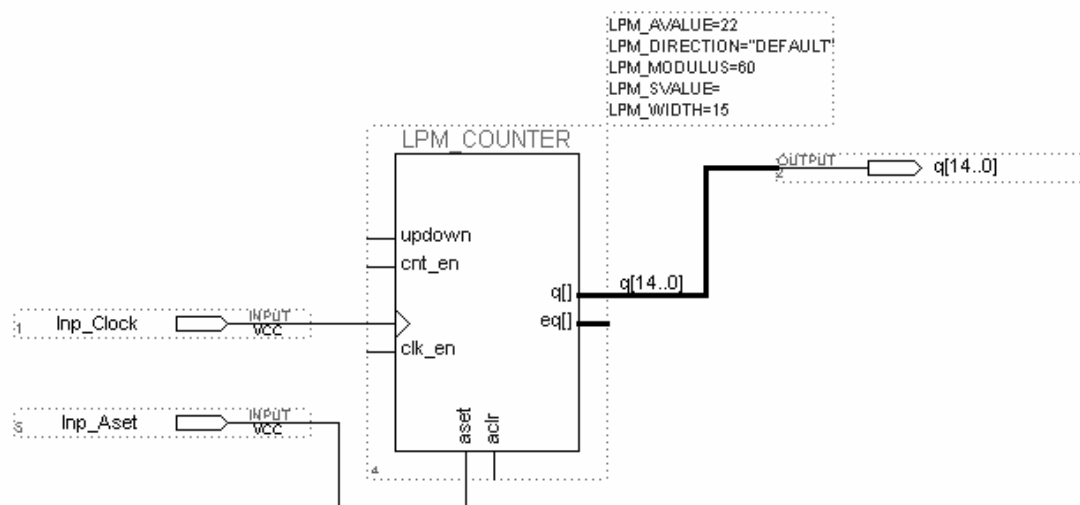
1. **q[]** – выходной многоразрядный код счётчика. Разрядность кода задаётся ниже величиной параметра **LPM_WIDTH**.
2. **eq[15..0]**
3. **cout** – выход переноса. На последнем такте счёта, т.е. на единицу меньшим числа равного модулю, вырабатывается импульс положительной полярности длительностью в один такт. Например, если модуль равен 60, то этот импульс будет на 60-ом такте (на временных диаграммах он будет занимать временную позицию под номером 59, т.к. первому такту соответствует номер 0).

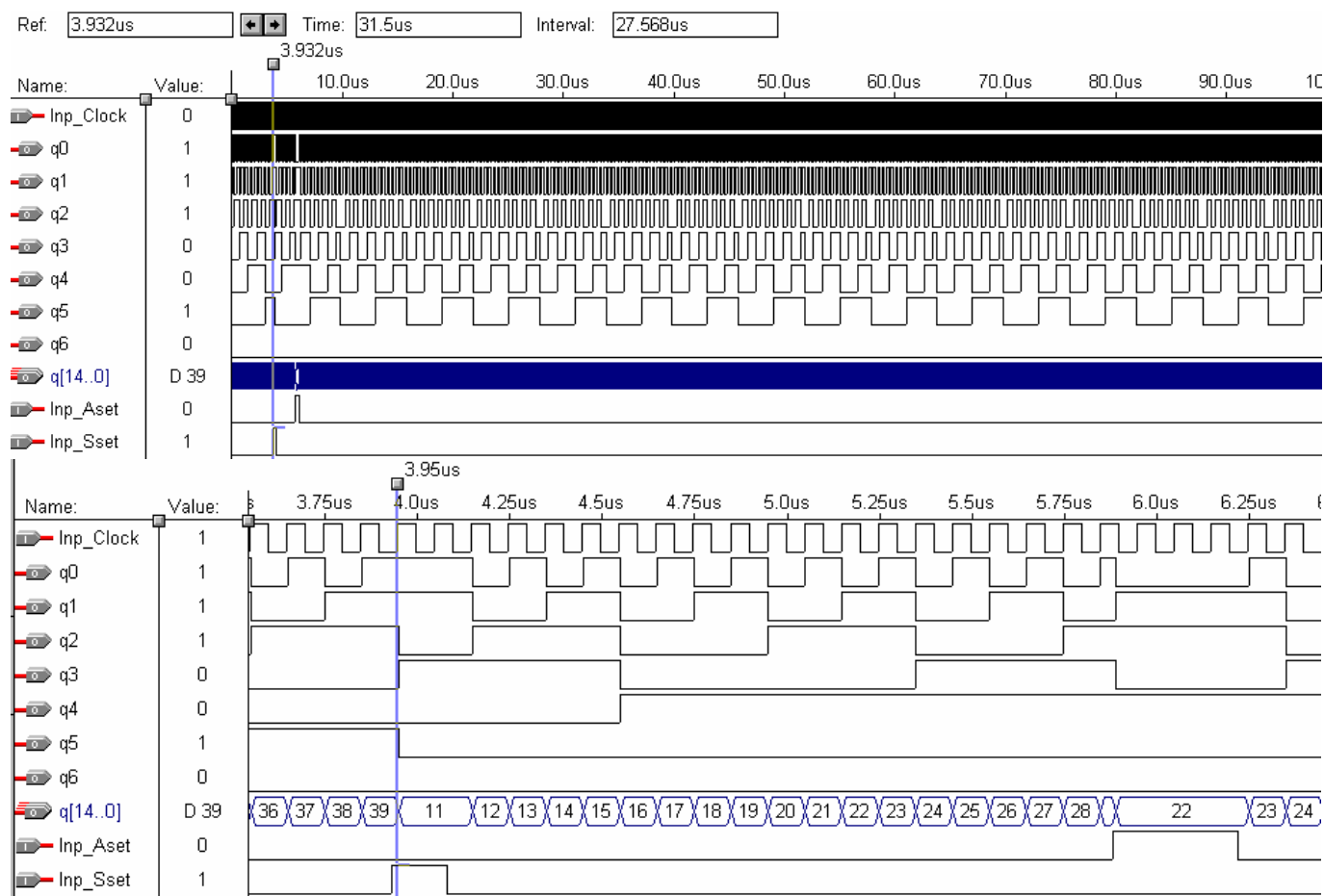
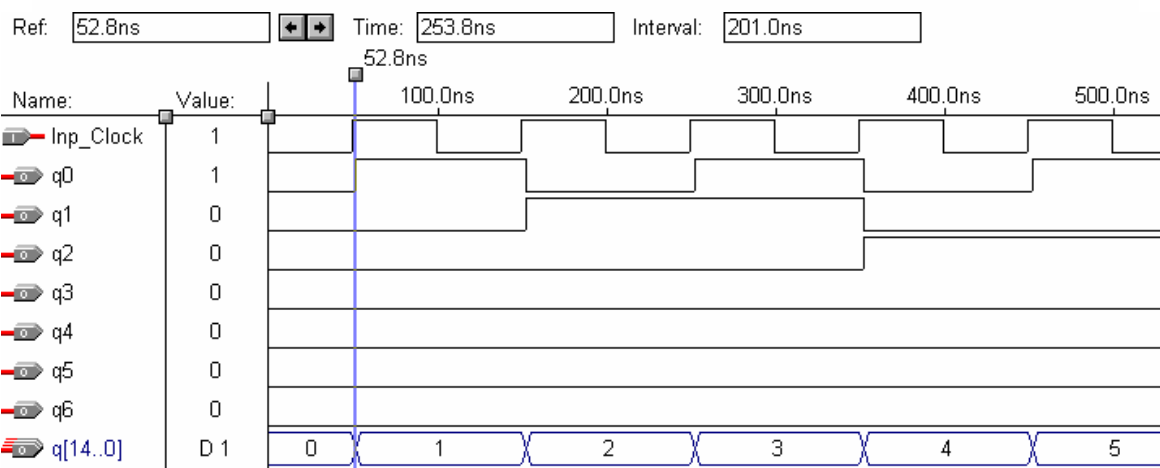
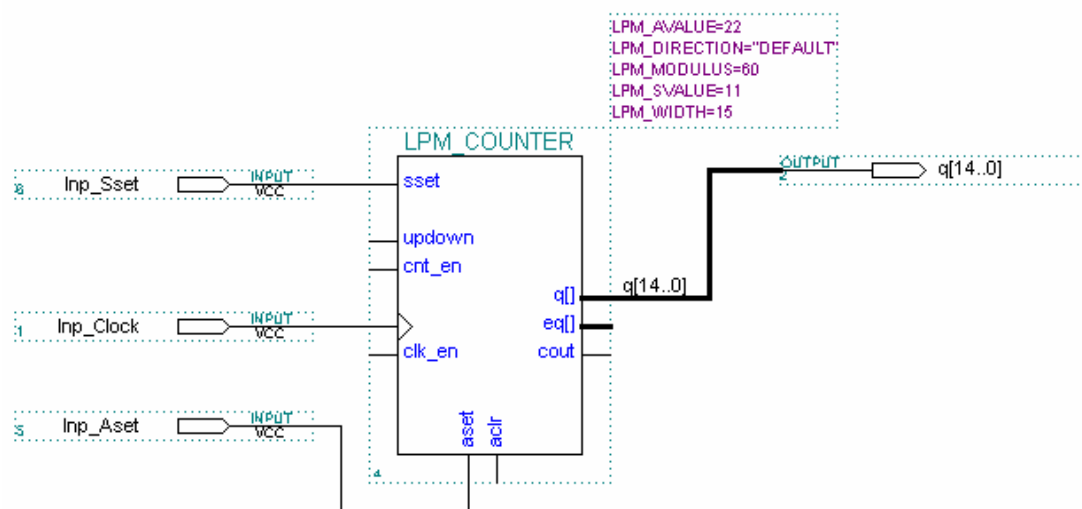
Параметры.

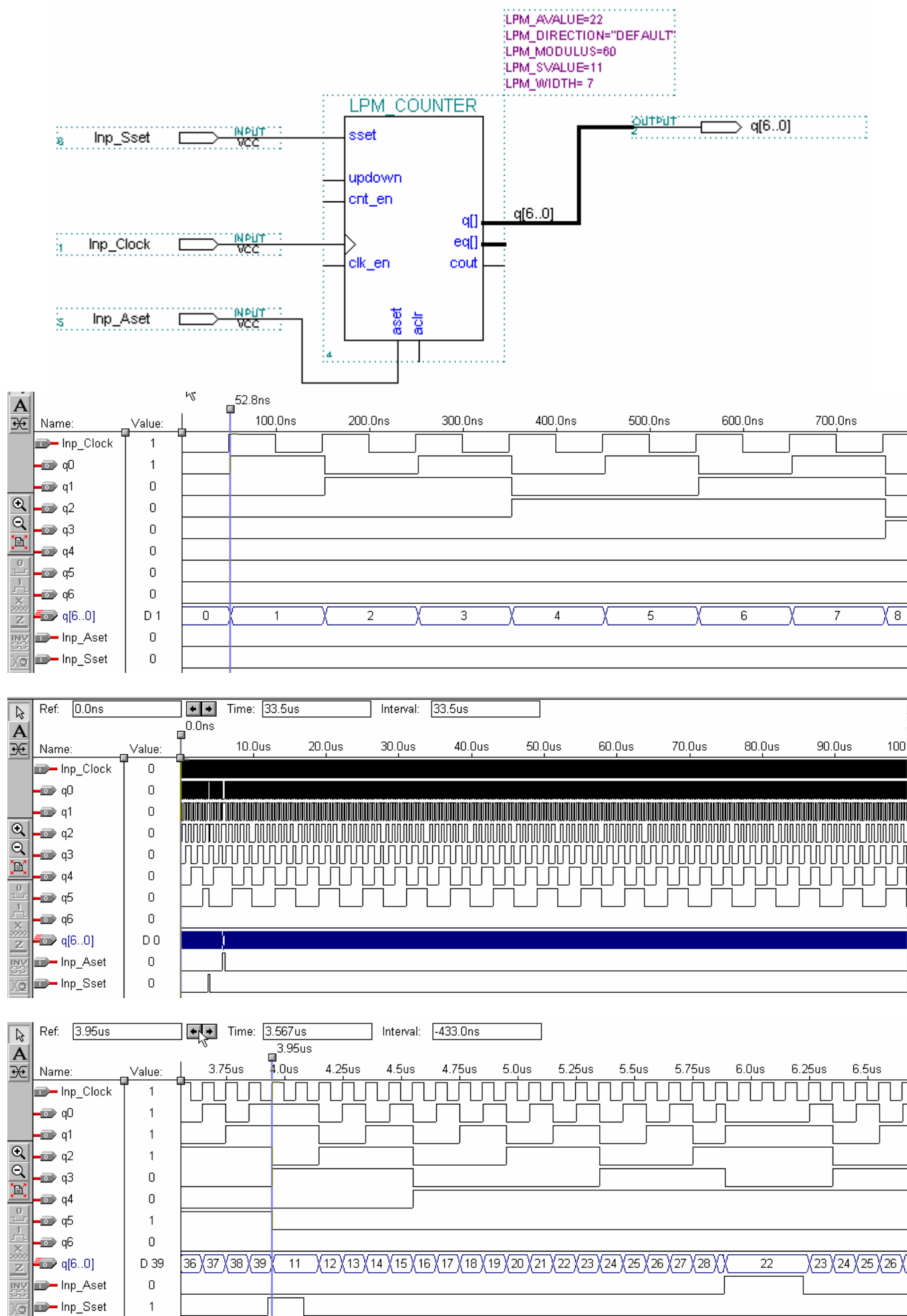
Параметры счетчика задаются в нижней части диалогового окна.



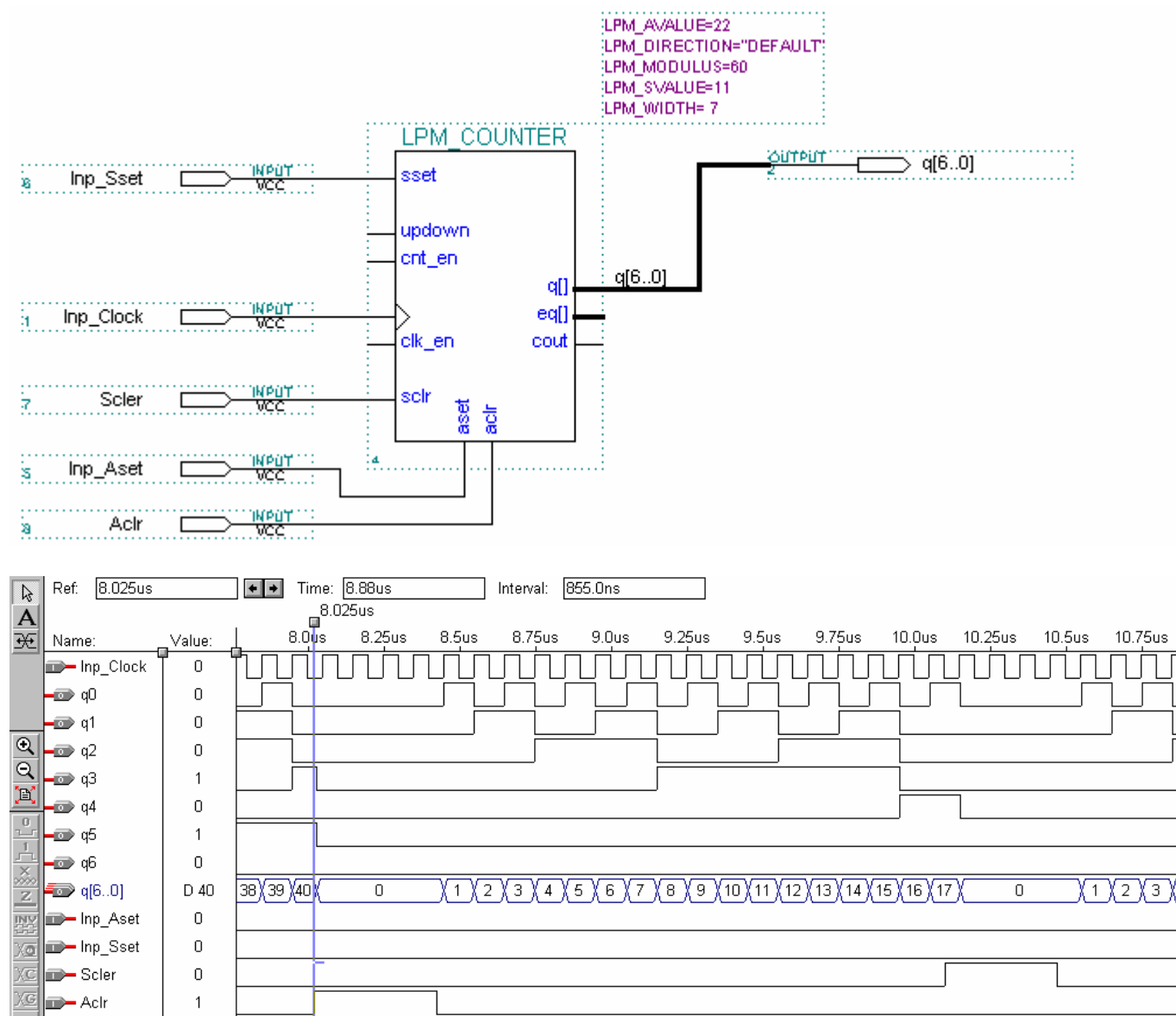
1. **LPM_AVALUE** – постоянная величина (константа), которая асинхронно передаётся на выход счётчика под действием уровня «1» на входе **aset**. Значение константы не должно превышать значения модуля, в противном случае состояние выхода счетчика будет неопределённым (**X**). Если вход **aset** не используется, то эта константа не воспринимается.
2. **LPM_DIRECTION** – выбираемая одна из трёх величина определяющая направление счёта: “UP”, “DOWN”, “DEFAULT”. При использовании этого параметра не следует подключать вход **updown**, при этом неопределённое значение параметра будет воспринято как “UP”.
3. **LPM_MODULUS** – Количество уникальных состояний выхода счетчика в одном цикле счёта.
4. **LPM_SVALUE** – вторая константа (постоянная величина), которая синхронно передаётся на выход счетчика под действием уровня «1» на входе **sset**.
5. **LPM_WIDTH** – количество разрядов в счётчике.



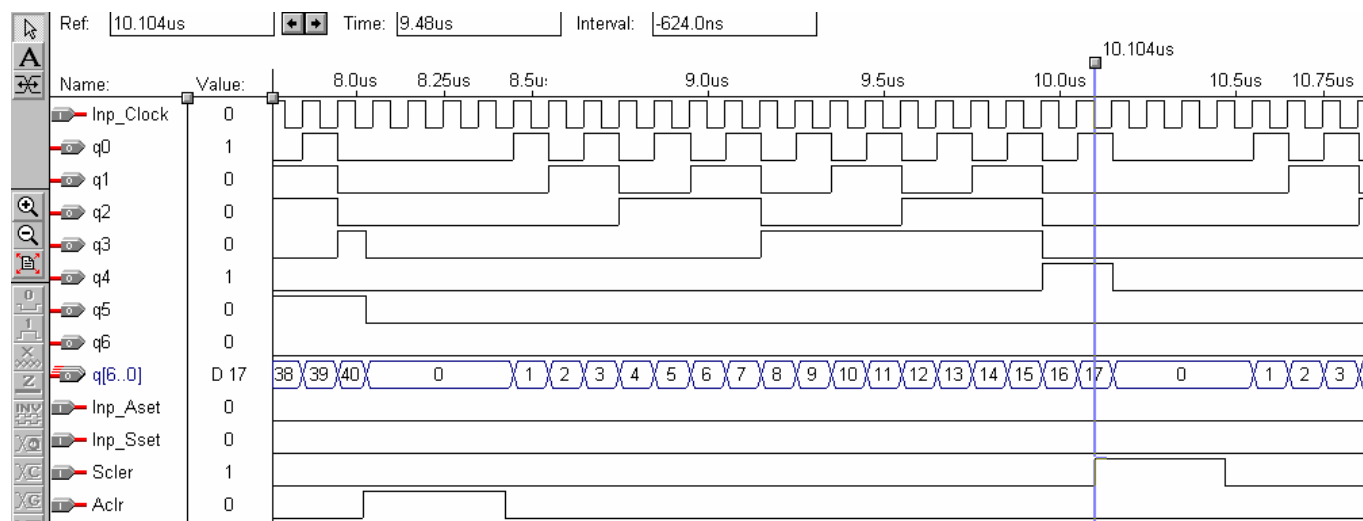




К использованию портов **aset**, **aclr**, **sset**, **sclr**.



Видно, что при асинхронном сбросе **aclr** обнуление счетчика не связано с синхрои́мпульсами **Clock**.



При синхронном сбросе **Sclr** начало и конец обнуления «привязаны» к ближайшим положительным перепадам синхрои́мпульсов **Clock**.

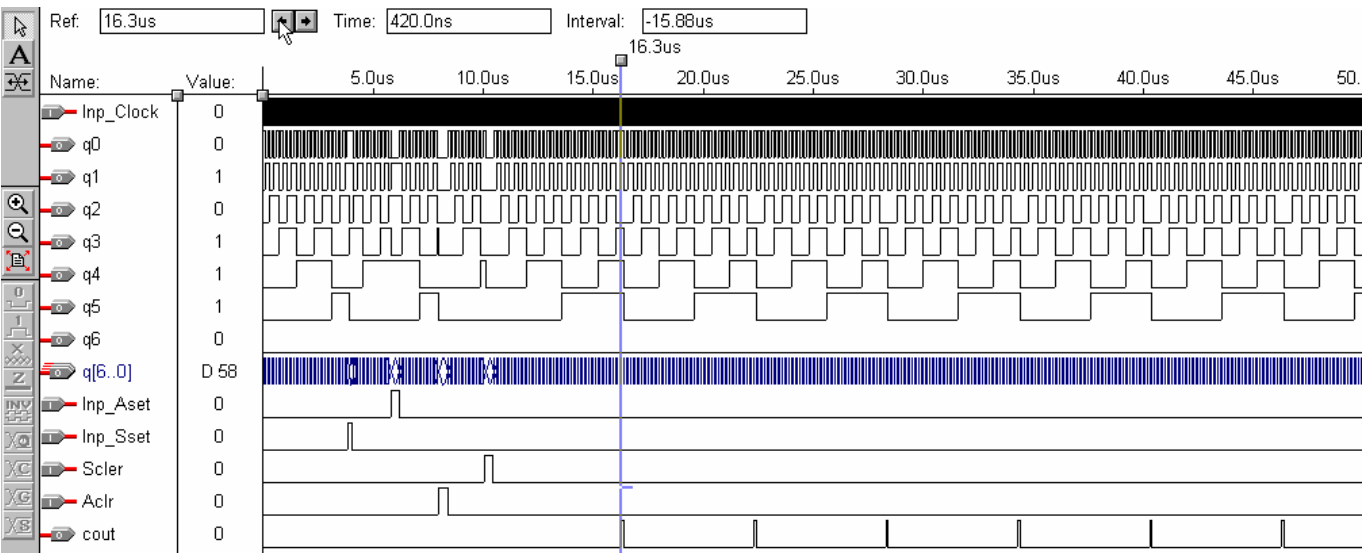
Таблица истинности для счётчика.

| Function: | | | | | | | | | | Function | |
|-----------|-------|--------|------|--------|--------|-------------------|---|---|---|------------|--|
| Inputs | | | | | | | | | | | Outputs |
| aclr | aload | clock | sset | cnt_en | | | | | | | |
| | aset | clk_en | sclr | sload | updown | q[LPM_WIDTH-1..0] | | | | | |
| 1 | X | X | X | X | X | X | X | X | X | 000... | Asynchronous set to value specified for LPM_AVALUE |
| 0 | 1 | X | X | X | X | X | X | X | X | 111... | |
| 0 | 1 | X | X | X | X | X | X | X | X | LPM_AVALUE | |
| 0 | 0 | 1 | X | X | X | X | X | X | X | data[] | Asynchronous load from data[] input |
| 0 | 0 | 0 | 0 | X | X | X | X | X | X | q[] | Hold current count value |
| 0 | 0 | 0 | 1 | ┐ | 1 | X | X | X | X | 000... | Synchronous Clear |
| 0 | 0 | 0 | 1 | ┐ | 0 | 1 | X | X | X | 111... | Synchronous set |
| 0 | 0 | 0 | 1 | ┐ | 0 | 1 | X | X | X | LPM_SVALUE | Synchronous set to value specified for LPM_SVALUE |
| 0 | 0 | 0 | 1 | ┐ | 0 | 0 | 0 | 0 | 0 | q[] | Hold current count value |
| 0 | 0 | 0 | 1 | ┐ | 0 | 0 | 1 | X | X | data[] | Synchronous load from data[] input |
| 0 | 0 | 0 | 1 | ┐ | 0 | 0 | 0 | 1 | 1 | q[]+1 | Count up |
| 0 | 0 | 0 | 1 | ┐ | 0 | 0 | 0 | 1 | 0 | q[]-1 | Count down |

Действие входов **aconst** и **sconst** в этой таблице не отражено. Из поисковой системы предположительно следует, что это эквивалентно значению **AVALUE** и вероятно **SVALUE** соответственно.

Ceiling of a real number:
CEIL(LOG2(255)) = 8
Floor of a real number:
FLOOR(LOG2(255)) = 7
Used status tested in an Assert Statement:
USED(aconst) == # 0 USED(AVALUE)

Временное положение выходных импульсов переноса.



Здесь больше 6 импульсов переноса **cout**. Причём, первый цикл счёта по модулю 60 занимает значительно больше тактов, чем 60. Это обусловлено тем, что напряжения на входах **asset**, **sset**, **aclr**, **Sclr** прерывают счёт на время своего действия и, соответственно, время достижения максимального числа равного модулю увеличивается. На следующих циклах прерываний нет, и импульсы переноса вырабатываются каждые 60 тактов (параметр **LPM_MODULUS** был задан 60)