



Счетчики.

Реализация на VHDL.



Счетчики.

Простой суммирующий счетчик.

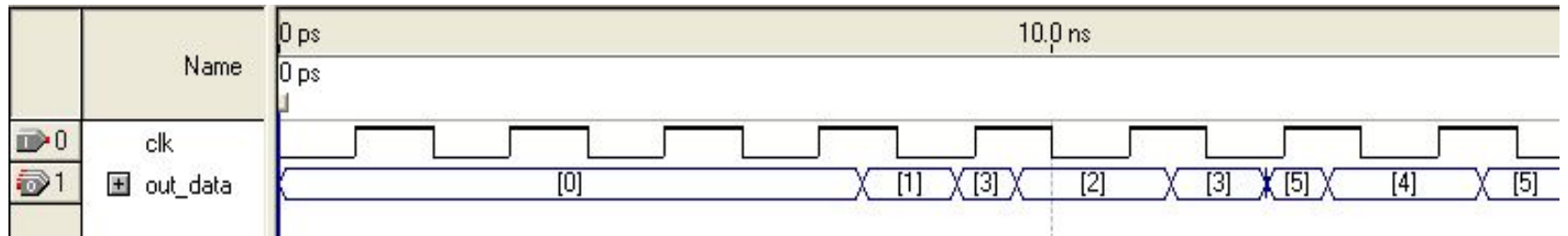
```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;
```

```
ENTITY count IS  
    PORT (clk: IN STD_LOGIC;  
          out_data: OUT INTEGER RANGE 0 TO 15);  
END count;
```

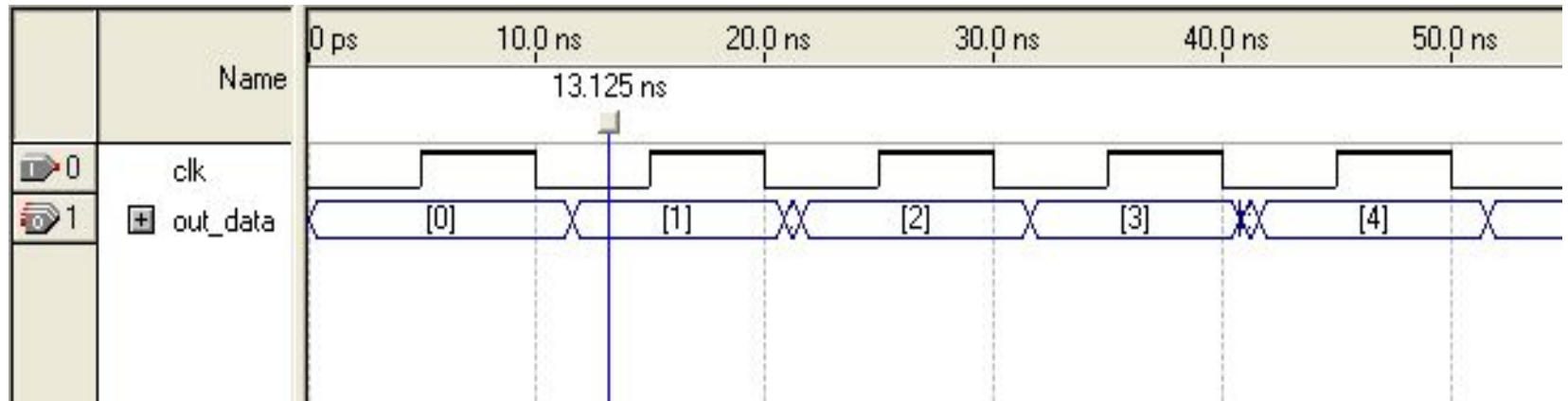
```
ARCHITECTURE a OF count IS  
-- внутренний сигнал, так как выход  
-- не может быть аргументом в выражении  
    SIGNAL csignal: INTEGER RANGE 0 TO 15;
```

```
BEGIN
  PROCESS (clk)
  BEGIN
    -- проверка наличия фронта
    IF (clk'EVENT AND clk = '1') THEN
    -- увеличение на единицу
    -- проверки на переполнение нет
      csignal <= csignal + 1;
    END IF;
  END PROCESS;
  out_data<= csignal;
END a;
```

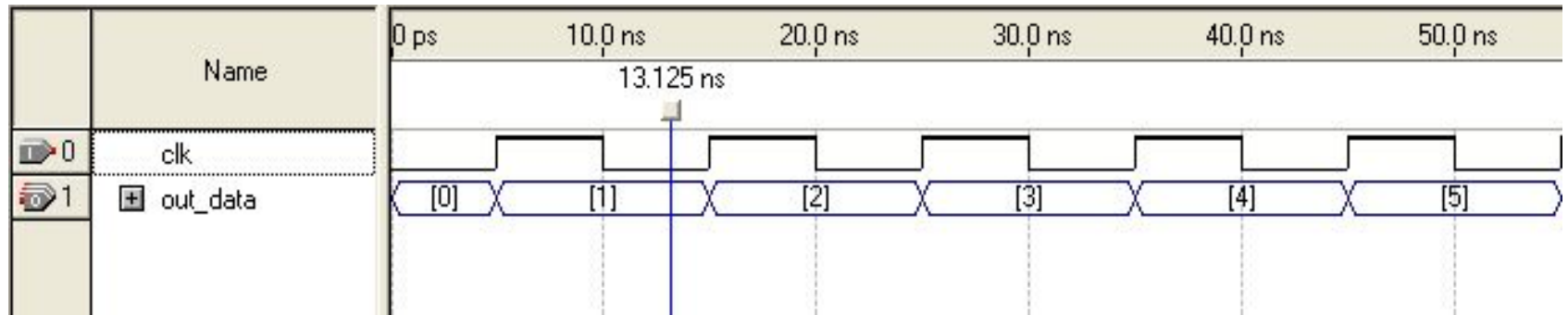
Временные диаграммы



Временные диаграммы



Диаграммы без учета задержек



Функциональная симуляция

- Это моделирование без учета задержек
- Для запуска:
 1. Выбрать функциональную симуляцию.
 2. Сгенерировать лист связей для функциональной симуляции (Functional Simulation Netlist)
 3. Запустить симуляцию

Выбор типа симуляции

Settings - count

Category:

- General
- Files
- Libraries
- Device
- + Operating Settings and Conditions
- + Compilation Process Settings
- + EDA Tool Settings
- + Analysis & Synthesis Settings
- Filter Settings
- + Timing Analysis Settings
- Assembler
- Design Assistant
- SignalTap II Logic Analy
- Logic Analyzer Interface
- [-] Simulator Settings
 - Simulation Verificatic
 - Simulation Output Fil
- PowerPlay Power Analyz
- SSN Analyzer

Simulator Settings

Select simulation options.

Simulation mode: Functional

Simulation input: Functional
Timing
Timing using Fast Timing Model

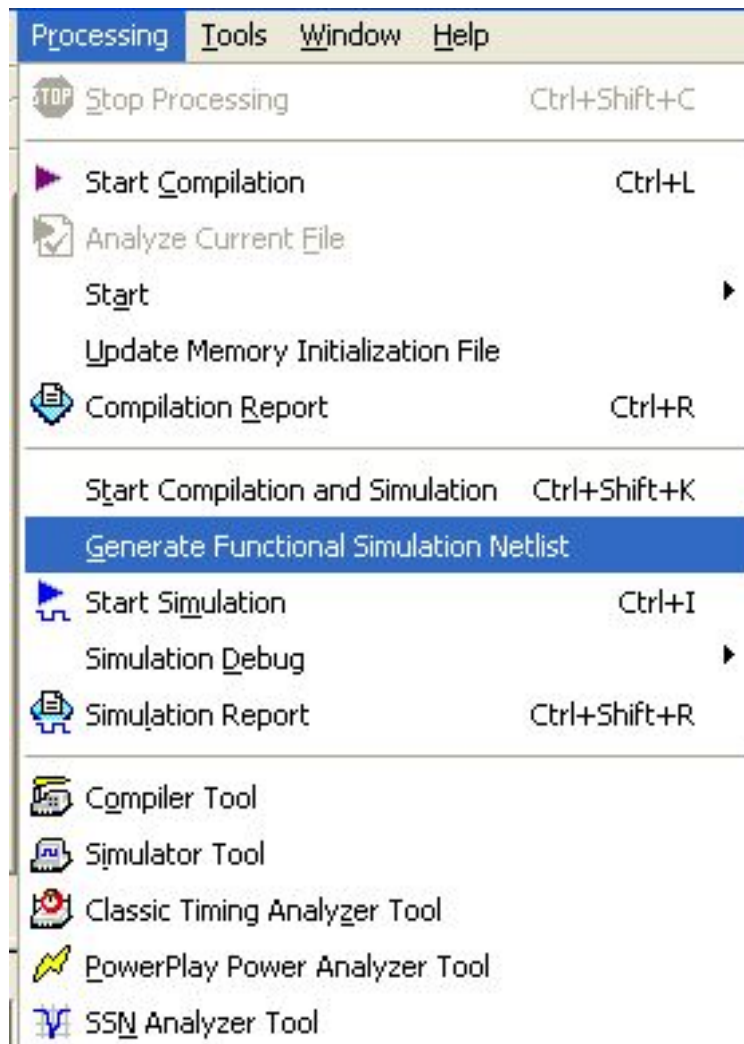
Simulation period

- Run simulation until all vector stimuli are used
- End simulation at: [] []

Simulation mode: Functional

Simulation input: Functional
Timing
Timing using Fast Timing Model

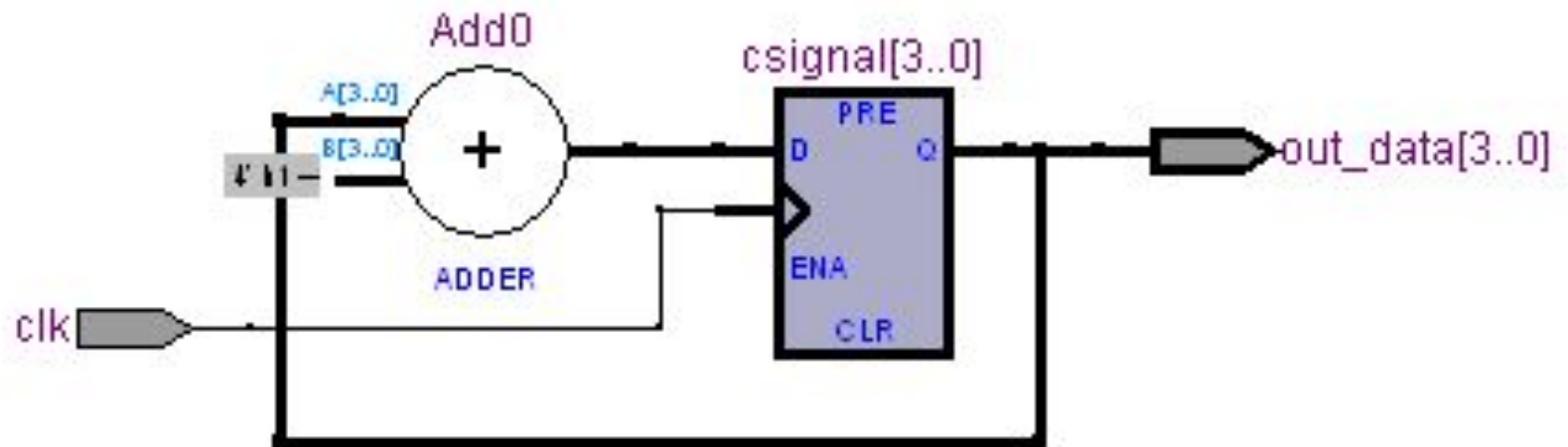
Генерация файла связей



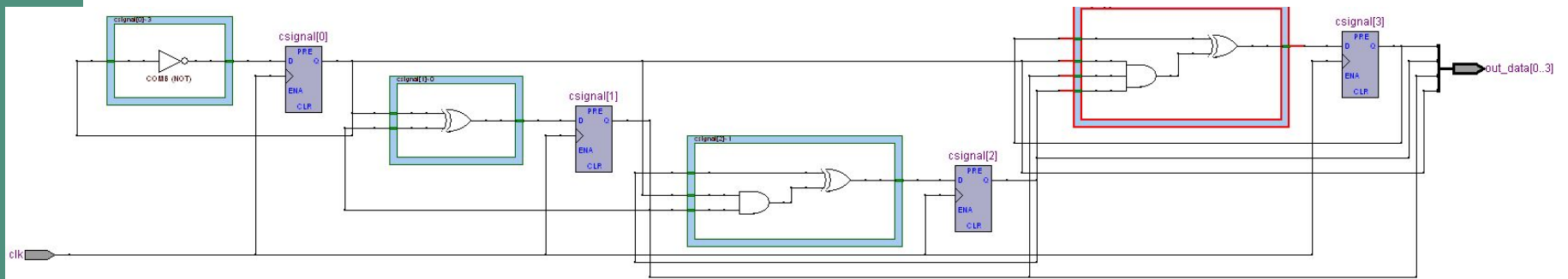
Меню:

Processing -> Generate
Functional Simulation
Netlist

Счетчик на RTL уровне



Счетчик на уровне ЛЭ ПЛИС



Реверсивный счетчик

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity count is
  generic (
    MIN_COUNT : natural := 0;
    MAX_COUNT : natural := 7);
  port (clk      : in std_logic;
        reset    : in std_logic;
        enable   : in std_logic;
        updown   : in std_logic;
        q       : out integer range MIN_COUNT to
        MAX_COUNT);
end entity;
```

```
architecture rtl of count is
    signal direction : integer;
begin
```

```
-- внутренний сигнал выбора направления счета
-- если сигнал равен 1 – сложение
-- «-1» - вычитание
```

```
process (updown)
begin
    if (updown = '1') then
        direction <= 1;
    else
        direction <= -1;
    end if;
end process;
```

```
process (clk)
  variable cnt : integer range MIN_COUNT to MAX_COUNT;
begin
  if (rising_edge(clk)) then
```

```
-- синхронный сброс
```

```
  if reset = '1' then
    cnt := 0;
```

```
-- проверка сигнала разрешения работы
```

```
  elsif enable = '1' then
```

```
-- счет
```

```
    cnt := cnt + direction;
```

```
  end if;
```

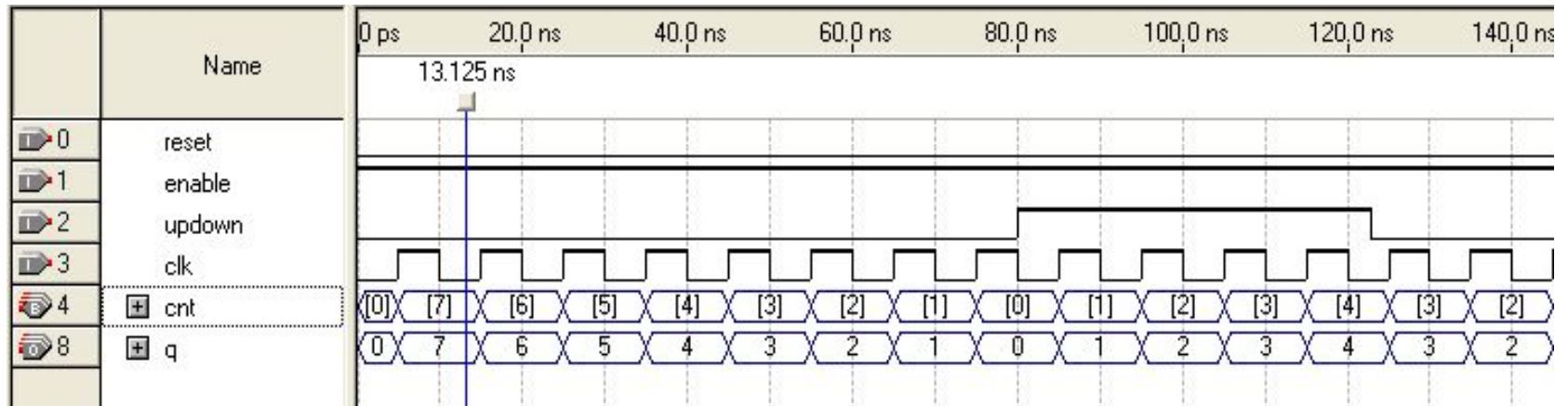
```
end if;
```

```
q <= cnt;
```

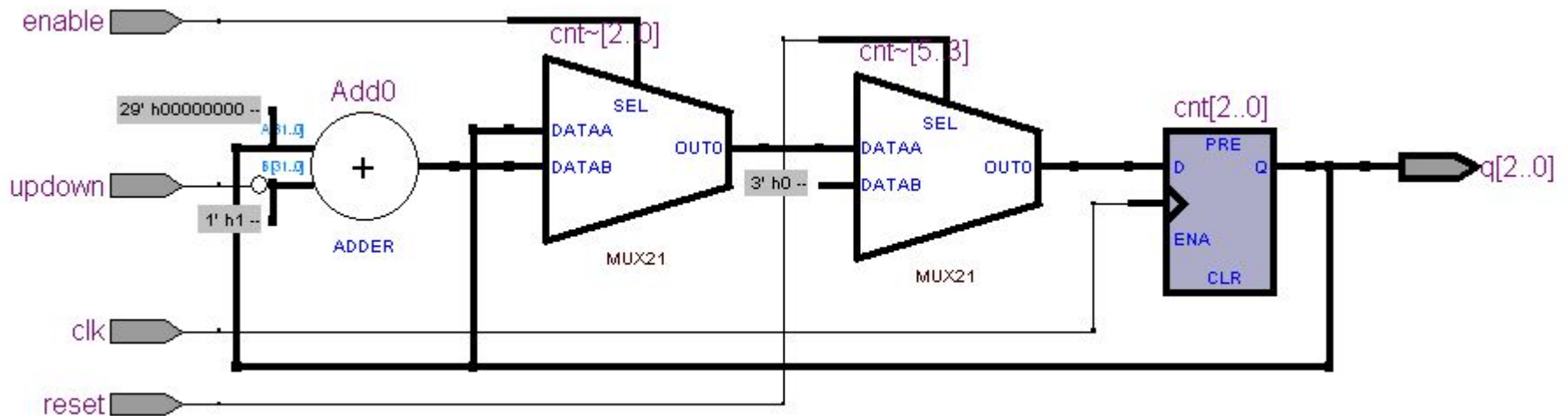
```
end process;
```

```
end rtl;
```

Временные диаграммы работы счетчика.



Счетчик на RTL уровне



Счетчик с асинхронным сбросом, синхронной предустановкой и сигналом разрешения счета.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY count IS
    PORT
        (d: IN INTEGER RANGE 0 TO 15;
         clk: IN STD_LOGIC;
         clrn: IN STD_LOGIC;
         ena: IN STD_LOGIC;
         load: IN STD_LOGIC;
         out_data: OUT INTEGER RANGE 0 TO 15
        );
END count;
ARCHITECTURE a OF count IS
    SIGNAL csignal: INTEGER RANGE 0 TO 15;
```

```
BEGIN
PROCESS (clk, clrn)
BEGIN
IF clrn = '0' THEN csignal <= 0;
ELSIF (clk'EVENT AND clk = '1') THEN
    IF load = '1' THEN csignal <= d;
    ELSE
        IF ena = '1' THEN
            csignal <= csignal + 1;
        ELSE
            csignal <= csignal;
        END IF;
    END IF;
END IF;
END PROCESS;
out_data<= csignal;
END a;
```

Счетчик.

Асинхронный сброс

-- clrн – сигнал сброса:

-- 0 – сброс, 1 – работа счетчика

IF clrн = '0' THEN csignal <= '0';

-- проверка на наличие фронта тактового

-- сигнала

ELSIF (clk'EVENT AND clk = '1') THEN

...

END IF;

Счетчик.

Синхронная загрузка

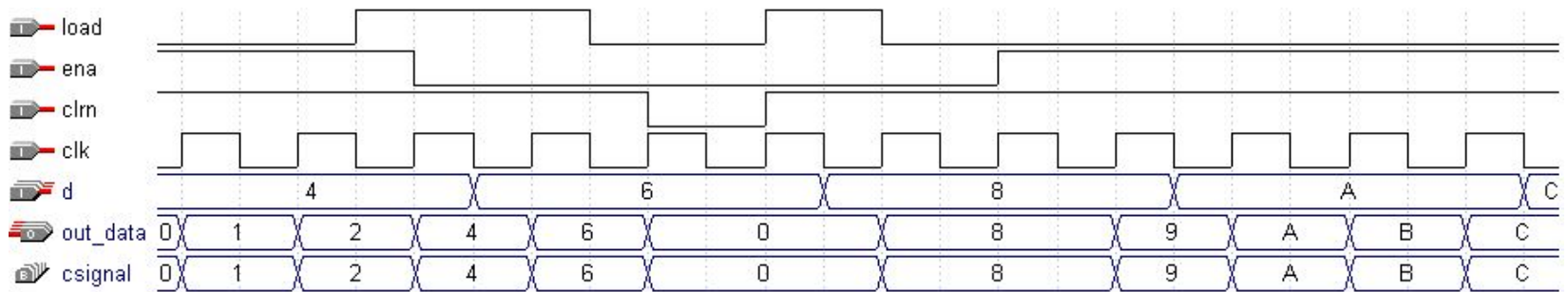
```
ELSIF (clk'EVENT AND clk = '1') THEN
  -- пришел фронт сигнала
  -- если активен сигнал загрузки load – запись
  -- входных данных в переменную csignal
  IF load = '1' THEN csignal <= d;
  ELSE
    IF ena = '1' THEN csignal <= csignal + 1;
    ELSE csignal <= csignal;
  END IF;
END IF;
```

Счетчик.

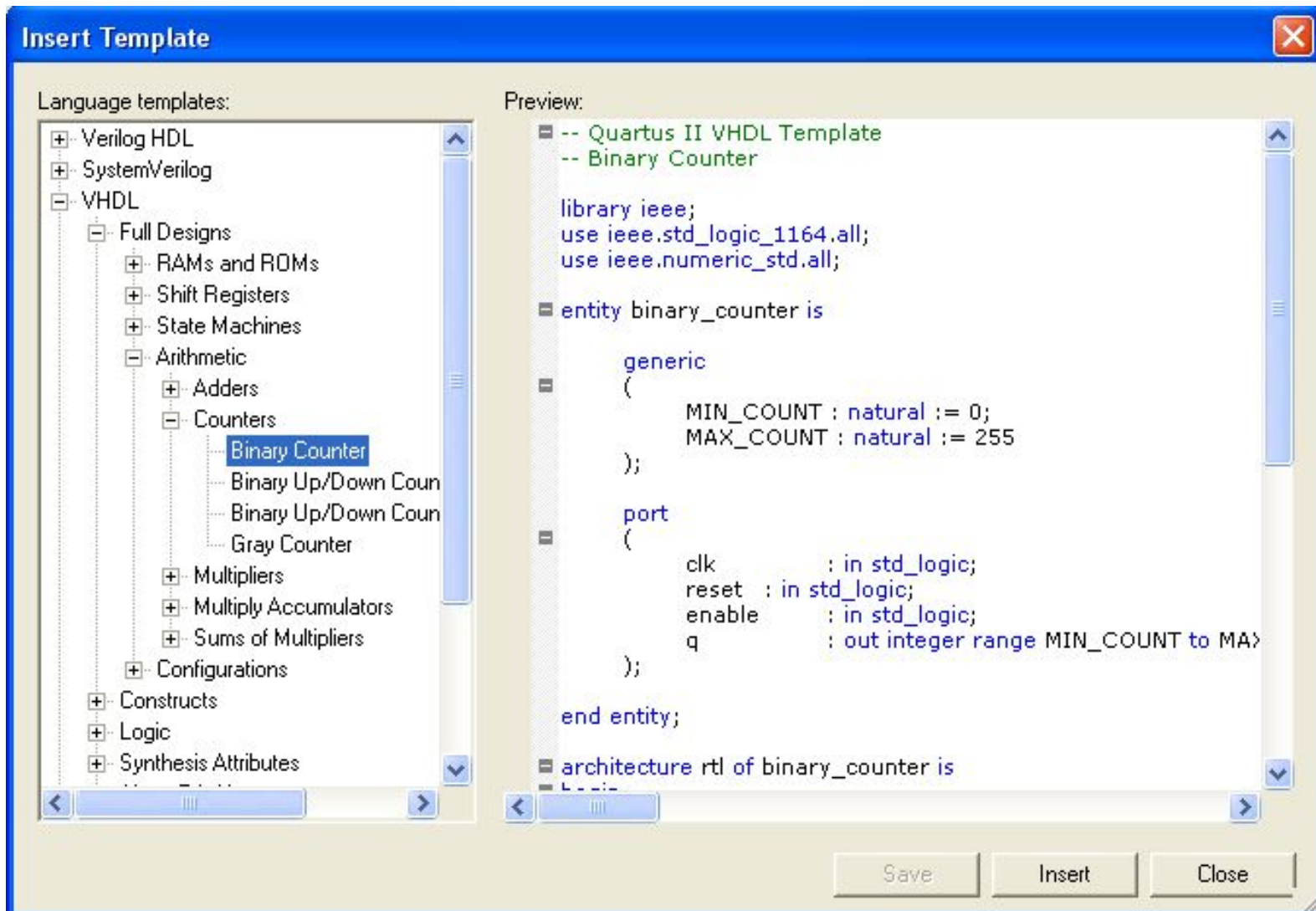
Синхронное разрешение счета.

```
IF ena = '1' THEN
  -- если активен сигнал разрешения ena
  -- то производится инкремент переменной
  csignal <= csignal + 1;
  ELSE   csignal <= csignal;
END IF;
```

Временные диаграммы работы счетчика.



Диалог шаблонов. ГОТОВЫЕ проекты



Дополнительная литература

- Quartus II Handbook. Vol. 1. Part 6. Recommended HDL Coding Styles.
- Xilinx HDL Coding Techniques.
http://www.xilinx.com/itp/3_1i/data/fise/xst/chap02/xst02000.htm

Внеклассное чтение. Идем в библиотеку

- <http://www.vokrugsveta.ru/vs/article/7353/>
- <http://travel.tochka.net/7531-samye-krasivye-biblioteki-mira-shest-must-visit/>
- <http://www.zotero.org/>
- <http://www.aquarium.ru/discography/biblioteka223.html>
- http://www.library.ru/lib/book.php?b_uid=42
- <http://ieeexplore.ieee.org/>
- <http://www.elsevier.com/>