

# Верификация разрабатываемого устройства

Верификация разрабатываемого устройства – это процесс, в ходе которого определяется, достигнуты или нет основные цели разработки устройства.

Верификацию можно разделить на :

- Формальную верификацию.
- Функциональную верификацию.

Формальная верификация выполняется без проведения моделирования работы устройства.

Функциональная верификация выполняет контроль устройства посредством моделирования его работы.

# Функциональная верификация

Основная задача проведения функциональной верификации — убедиться в том, что разработанное устройство полностью реализует требуемые функции.

При проведении функциональной верификации тестируемое устройство может рассматриваться как :

- «черный ящик»
- «белый ящик»
- «серый ящик»

# Моделирование

Моделирование – воспроизведение реального поведения устройств с заданной степенью точности при помощи последовательных алгоритмов, реализуемых в ЭВМ.

Особенности моделирования устройств:

- Для выполнения моделирования работы устройства необходимо наличие тестовых воздействий, подаваемых на его входы.
- Анализ корректности работы устройства выполняется внешними, по отношению к инструменту моделирования и тестируемому устройству, объектами.

Два основных подхода к построению систем моделирования дискретных устройств:

- сквозное моделирование ;
- событийное моделирование .

# Сквозное моделирование

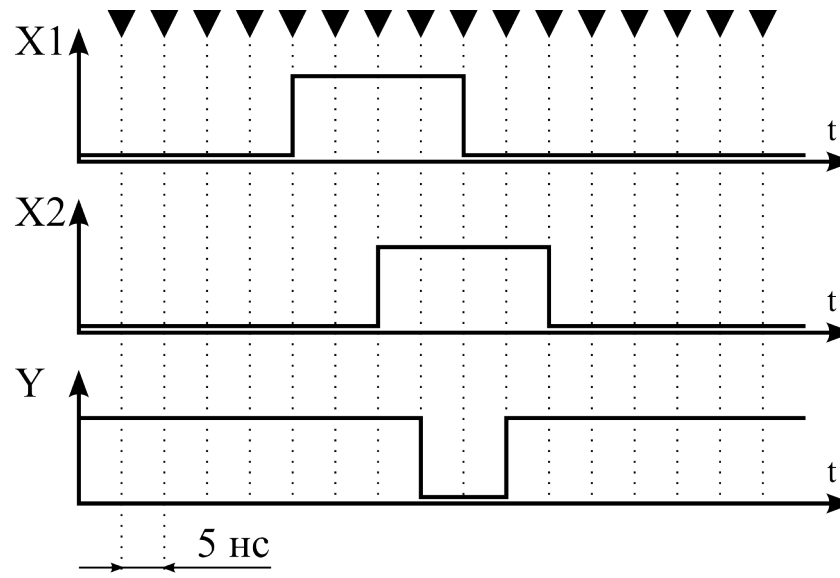
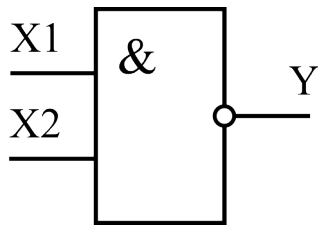
При сквозном моделировании время моделирования делится на кванты. Каждый квант реального времени соответствует единице модельного времени и отображается одним шагом моделирования.

Основной недостаток — нерациональные затраты машинного времени, вызванные потребностью достаточно мелкой дискретизации времени и необходимостью воспроизведения на каждом шаге поведения всех компонентов, в том числе тех, на входах которых не происходит изменение сигналов.

# Пример реализации сквозного моделирования

Задержка переключения логического элемента – 5 нс.

Квант времени моделирования (шаг моделирования) – 5 нс.



# Событийное моделирование

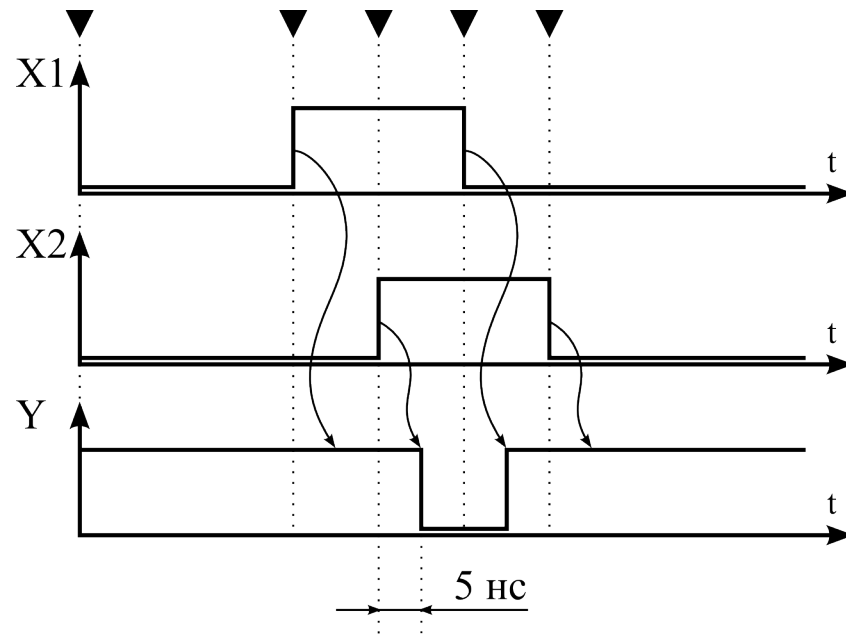
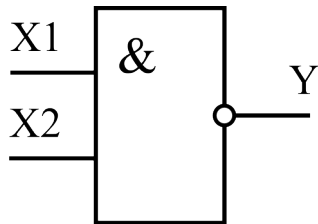
При событийном моделировании на каждом шаге моделирования заново вычисляются состояния только тех компонентов, на входе которых в данный момент происходят изменения.

Изменения на входах называют *событиями*. Любое *событие* может вызвать цепочку событий.

Моделирование системы производится не для равномерно отстоящих моментов реального времени, а лишь для моментов, для которых ранее были предсказаны *события*.

# Пример реализации событийного моделирования

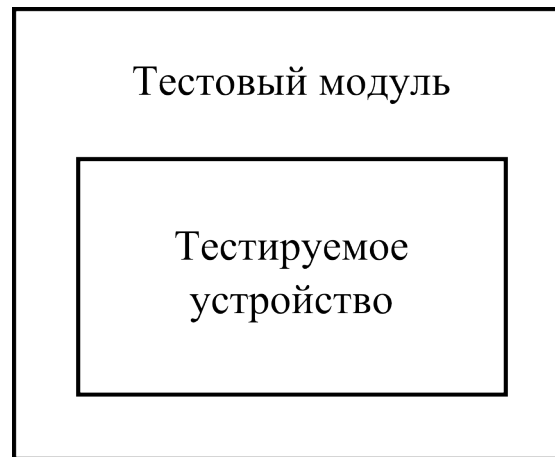
Задержка переключения логического элемента – 5 нс.



# Тестовый модуль (Testbench)

Термин **Testbench** (*тестовый модуль*) чаще всего используется для обозначения кода, предназначенного для подачи на тестируемое устройство определенного тестового воздействия и , при необходимости, считывания и обработки результатов работы устройства по этому воздействию.

Взаимосвязь между тестовым модулем и тестируемым устройством :





# Тестовый модуль (Testbench)

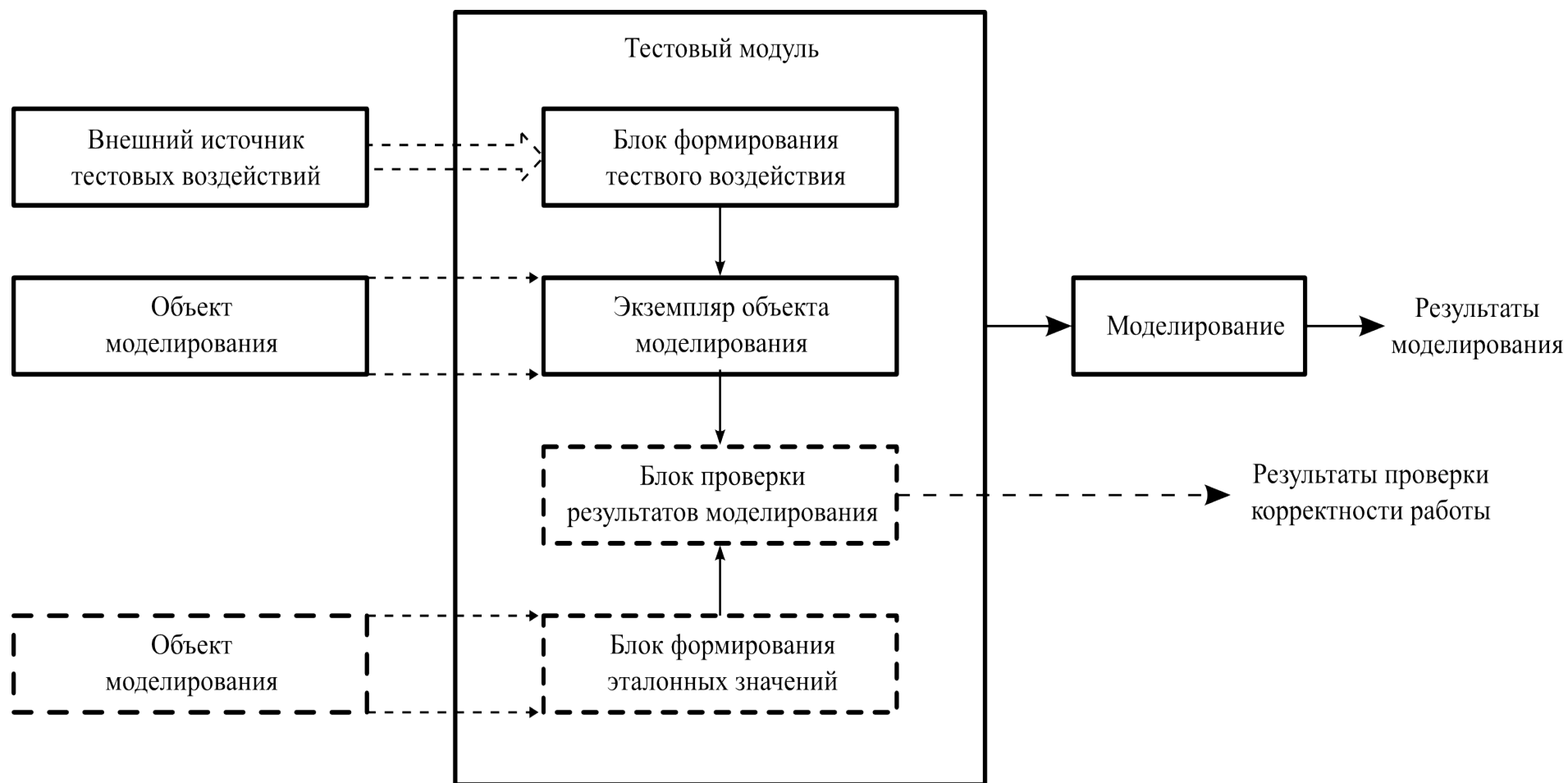
Основные задачи, решаемые тестовым модулем :

- Подача входных тестовых воздействий на тестируемое устройство.
- Анализ функционирования устройства на заданных тестовых воздействиях.

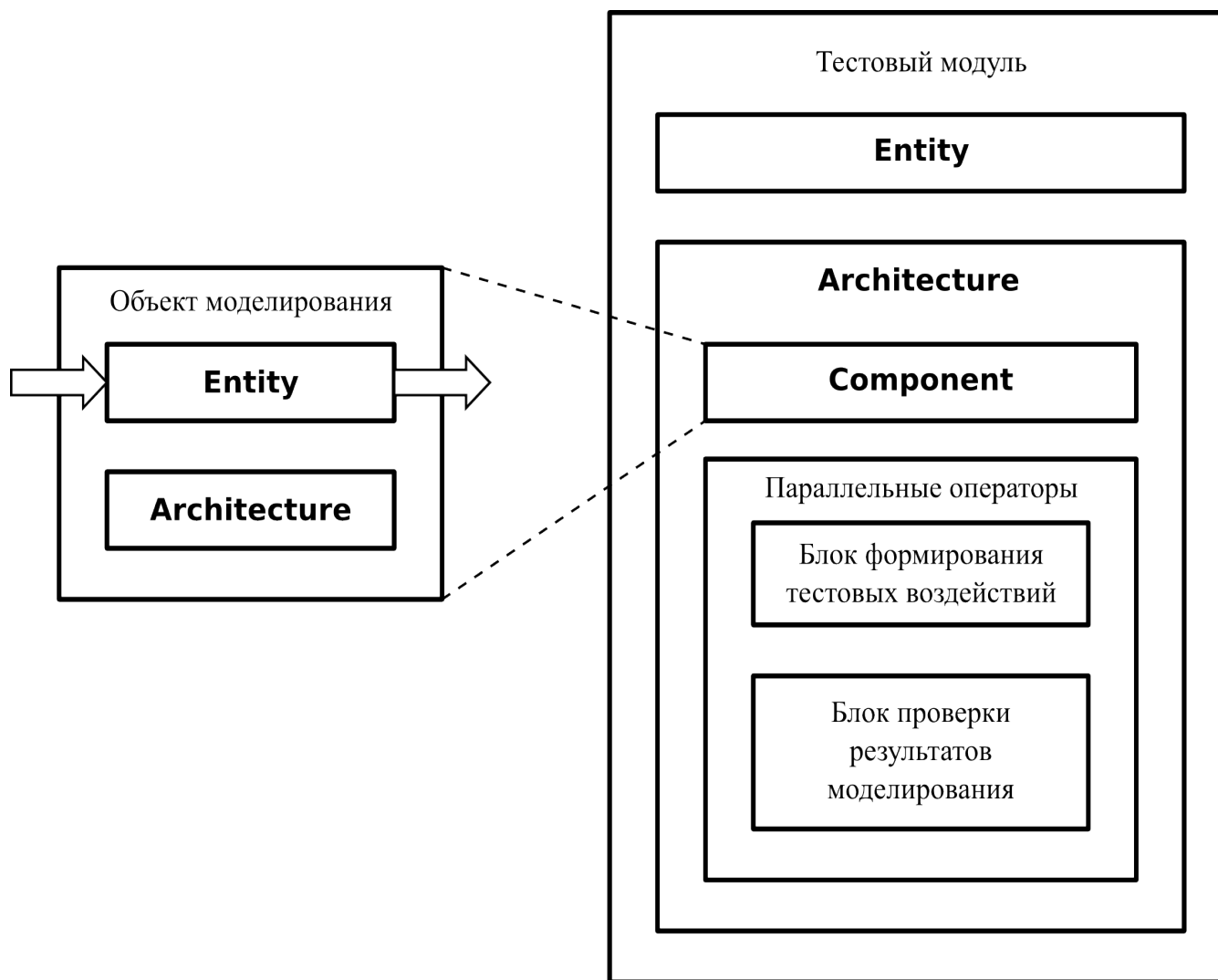
Анализ функционирования устройства выполняется путем сравнения результатов работы устройства с эталонным значением.

Методы получения тестовых воздействий и эталонных значений могут быть различны.

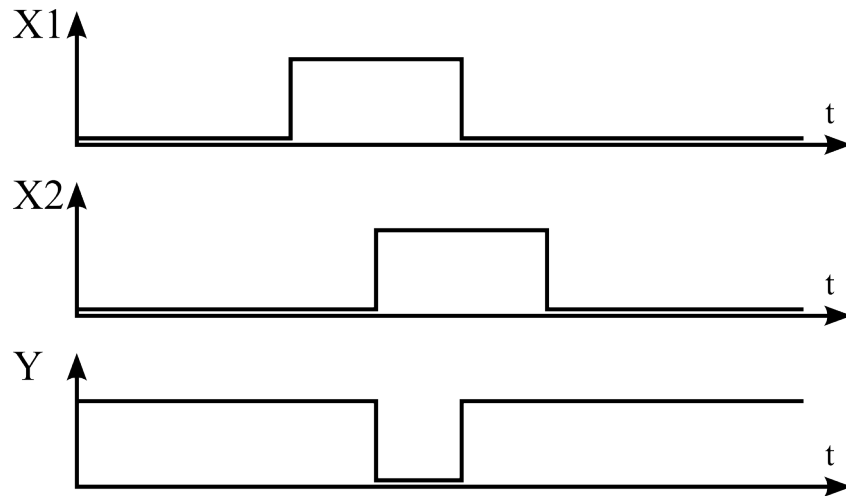
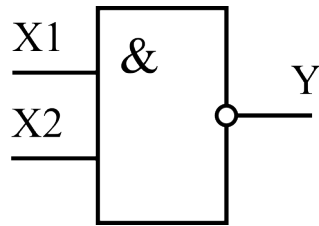
# Общая структура тестового модуля



# Тестовый модуль на языке VHDL

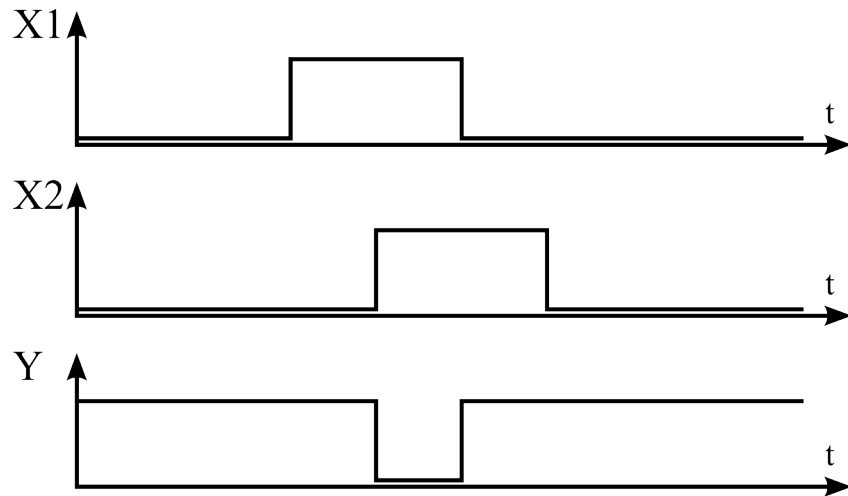
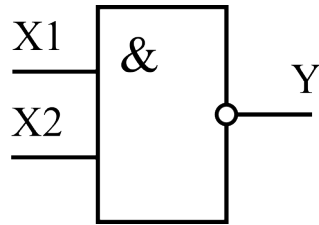


# Пример тестового модуля на языке VHDL



```
entity test_bench is
end test_bench;
architecture test_bench of test_bench is
    component not_and
        port (
            x    : in std_logic_vector(1 downto 0);
            y    : out std_logic);
        end component;
    signal x      : std_logic_vector(1 downto 0);
    signal y      : std_logic;
begin
    uut : test_bench port map (x => x, y => y);
    process
    begin
        for i in 0 to 3 loop
            x <= conv_std_logic_vector(i);
            wait for 10 ns;
        end loop;
        assert false
            report "End simulation"
            severity failure;
    end process;
end test_bench
```

# Пример тестового модуля на языке VHDL (блок проверки корректности работы)



```
...  
process (x)  
    variable temp : std_logic;  
begin  
    temp := not (x(1) and x(2));  
    assert y = temp  
        report "Error"  
        severity failure;  
end process;  
...
```

