

# Помощь студенту к первой лабораторной работе

---

ПУМУ 2 СЕМЕСТР

---

## **ПРЕДУПРЕЖДЕНИЕ!**

Данные лабораторные работы достаточно **сложны для понимания**, в презентации **ПОДРОБНО описаны** ключевые моменты. На выполнение лабораторной работы даётся всего **ОДНА пара**, поэтому Ваша задача – **не сидеть с открытым пустым проектом** половину занятия (как это обычно бывает), а сразу **втягиваться в процесс** и делать всё по пунктам.

---

**ОТКРЫВАЙТЕ МЕТОДИЧКУ (22 стр) И  
СОЗДАВАЙТЕ ПРОЕКТ СОВМЕСТНО С  
МАТЕРИАЛАМИ ИЗ ПРЕЗЕНТАЦИИ**

**Если возникают **несостыковки**, смотрите сюда**

# Лабораторная работа №1 (стр. 2



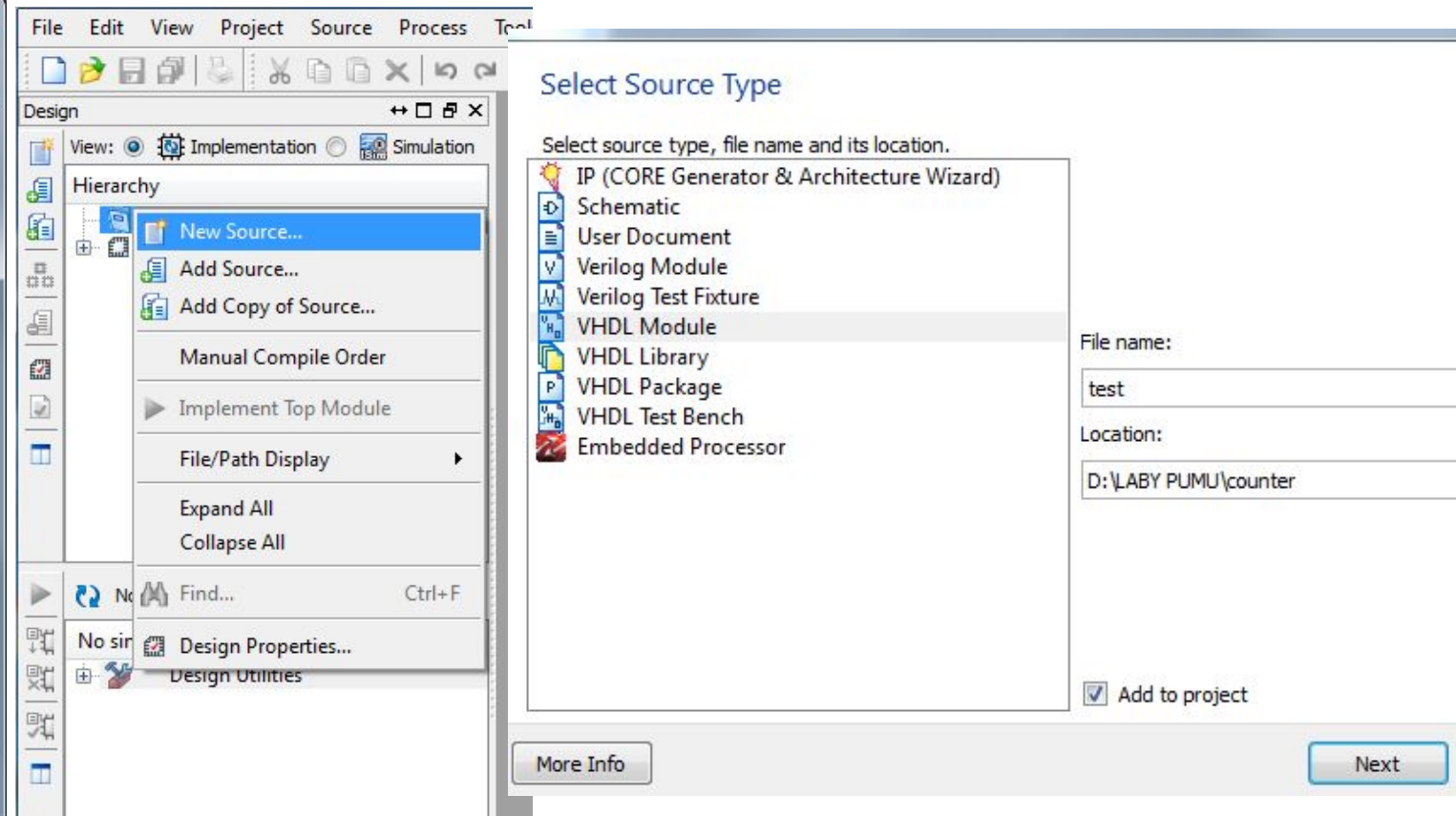
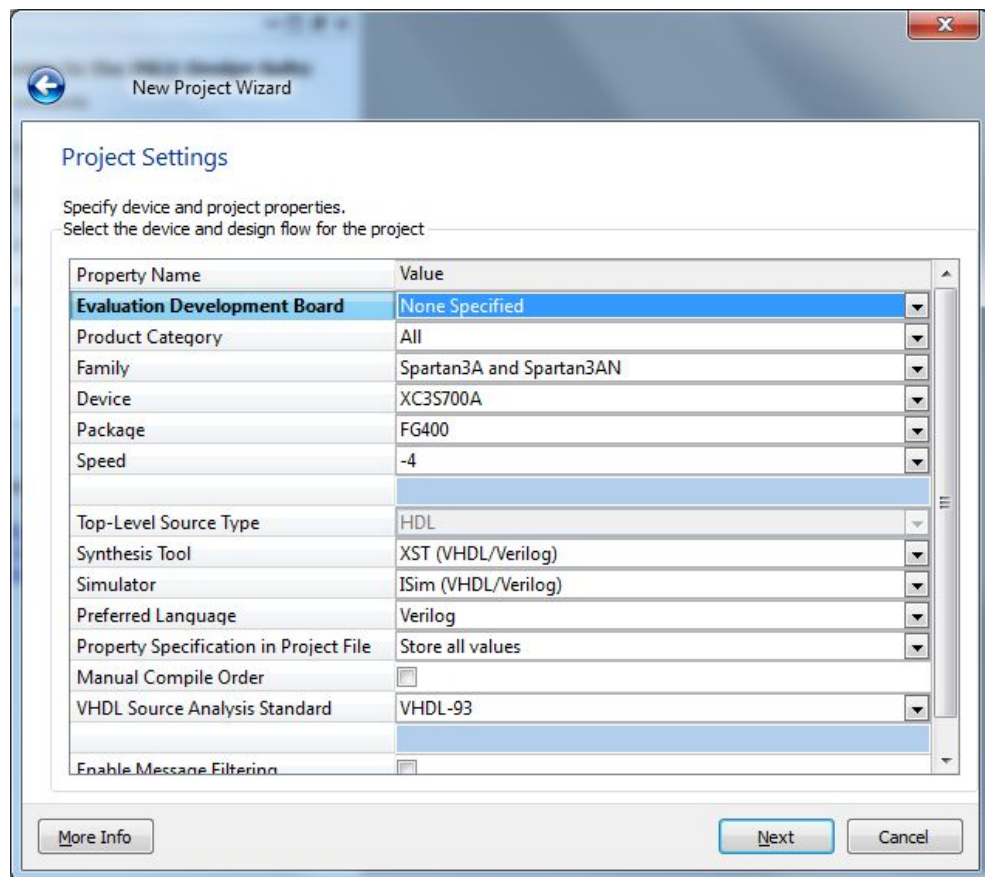
При выполнении лабораторных работ используется программа **ISE Project Navigator**. На компьютерах установлена новая версия программы, а методичка описывает старую версию. В данной памятке отмечены те пункты, которые различаются с методическим пособием. В остальных лабораторных работах (2,3,4) Вы будете самостоятельно разрабатывать устройства, поэтому **ВНИМАТЕЛЬНО** ознакомьтесь с алгоритмом разработки и анализа.

P.S.

Защита лабораторных работ в этом семестре будет происходить **БЕЗ ОТХОДА** от преподавателя (за исключением специальных сложных вопросов). Задаются вопросы – не ответили, приходите в следующий раз. Отвечающему даётся на ответ **7 минут**.

# Отличия от методического пособия

## 1.2 пункт 7 - пропускаем



---

✘ `unter/test.vhd`" Line 46. + can not have such operands in this context.  
✘ `unter/test.vhd`" Line 48. - can not have such operands in this context.

---

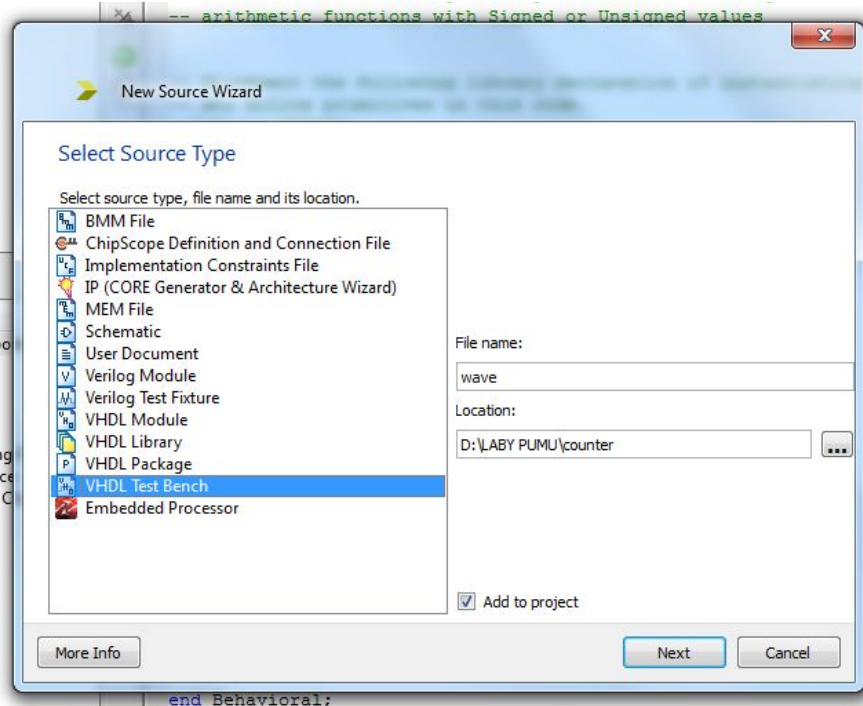
Может возникнуть **ошибка**. Чтобы её исправить, необходимо добавить строчку

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use ieee.std_logic_unsigned.all;
```

**ieee.std\_logic\_unsigned.all**

(значение строки спрощу у Вас на защите!!!)

# В 1.4.1 пункте 3 выбираем **VHDL Test Bench** (модуля из методички нет в новой версии)



```
-- arithmetic functions with Signed or Unsigned values
signal COUNT_OUT : std_logic_vector(3 downto 0);

-- Clock period definitions
constant CLOCK_period : time := 10 ns;

EGIN

-- Instantiate the Unit Under Test (UUT)
uut: test PORT MAP (
    CLOCK => CLOCK,
    DIRECTION => DIRECTION,
    COUNT_OUT => COUNT_OUT
);

-- Clock process definitions
CLOCK_process :process
begin
    CLOCK <= '0';
    wait for CLOCK_period/2;
    CLOCK <= '1';
    wait for CLOCK_period/2;
end process;

-- Stimulus process
stim_proc: process
begin
    -- hold reset state for 100 ns.
    wait for 100 ns;

    wait for CLOCK_period*10;

    -- insert stimulus here
```

Настройка времени тактирующего сигнала, сигнала, отвечающего за направление счета происходит **ТЕКСТОВО**. Т.е. тут нет графического описания, а есть код, который Вам необходимо подкорректировать по заданным параметрам.

Соответственно пропускаем 1.4.1 после 7го пункта, **БЕРЕМ ОТ ТУДА ТОЛЬКО ПАРАМЕТРЫ ТРЕБУЕМОГО СИГНАЛА:**

**ПЕРИОД, ВРЕМЯ «0», ВРЕМЯ «1», ОБЩЕЕ ВРЕМЯ СИГНАЛА**

# ПОСМОТРИТЕ ВНИМАТЕЛЬНО НА КОД

---

```
-- Clock period definitions
constant CLOCK_period : time := 40 ns;
```

```
-- Clock process definitions
CLOCK_process :process
begin
    CLOCK <= '0';
    wait for CLOCK_period/2;
    CLOCK <= '1';
    wait for CLOCK_period/2;
end process;
```

Подумайте, ЧТО описывают эти строки, **КАК** нужно их **отредактировать** для получения необходимых параметров?

```
begin
    -- hold reset state for 100 ns.
    wait for 100 ns;

    wait for CLOCK_period*10;

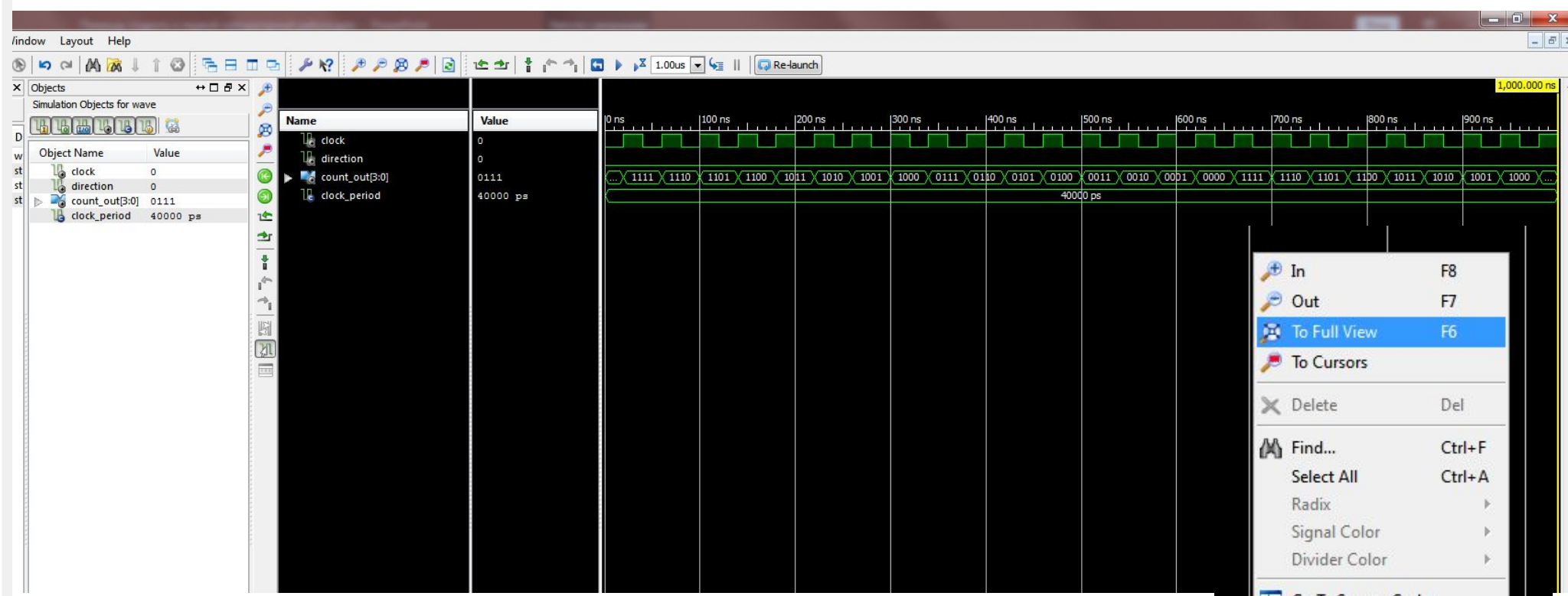
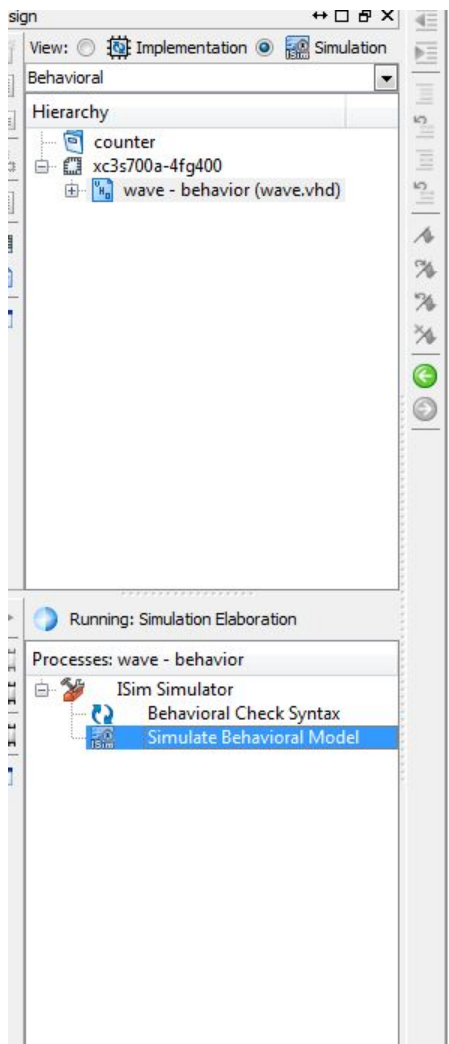
    -- insert stimulus here

    wait;
```



# Переходим к Функциональному моделированию

## 1.4.3



Для запуска моделирования выбираем **Simulate Behavioral Model**

Обращайте внимание на время, выставленное в окне моделирования, так же воспользуйтесь функцией **To Full View** для того, чтобы масштабировать временные характеристики

- In F8
- Out F7
- To Full View F6**
- To Cursors
- Delete Del
- Find... Ctrl+F
- Select All Ctrl+A
- Radix >
- Signal Color >
- Divider Color >
- Go To Source Code
- Show Drivers
- Force Constant...
- Force Clock...
- Remove Force
- Go To... Ctrl+G
- Cursors >
- Markers >
- New Divider

---

**ПУНКТЫ, НАЧИНАЯ С 1.5 НЕ ДЕЛАЮТСЯ**

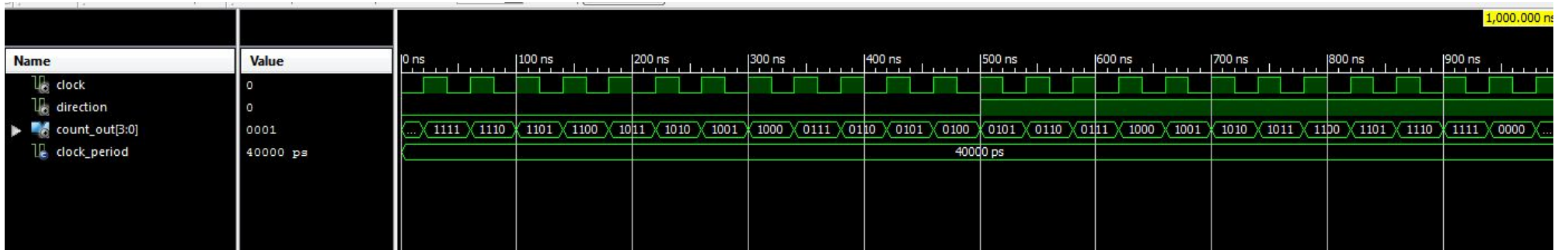
Задания на следующем слайде

# Задания

1. Смените направление счёта.
2. Досчитать до половины счёта и сменить направление.

В отчёт вставляете код файла VHDL описания, а так же код файла для параметров симуляции двух заданий и временные характеристики (не забудьте развернуть count\_out).

Пример временных диаграмм со сменой направления счета:



---

# УДАЧИ С ЛАБОРАТОРНОЙ РАБОТОЙ!

- Встретимся на защите ;)