
Simtera. Новости продукта и новый функционал

//

Добрый день!

Сегодня мы начинаем цикл рассказов о HDL- симуляторе Simtera, входящим в систему Delta Design. За прошедший год Simtera шагнула вперед и нам есть о чем рассказать.

Что такое Simtera?

Что же такое HDL - симулятор? HDL, или Hardware Description Language - язык описания аппаратуры. Широко распространены два языка описания - Verilog и VHDL. Они создавались в 80-ых годах прошлого столетия для описания *логических* схем. Нужно было, чтобы схемы строились иерархически и складывались в блоки, модули и системы. Такая иерархия легко поддается моделированию на любом из уровней. HDL-языки позволяют не только описывать работу логических схем, но и на них создается программное обеспечение для ПЛИС, благодаря синтезируемому подмножеству языков. Simtera является удобным инструментом для проверки кода на наличие ошибок перед "прошивкой" интегральных схем, позволяет моделировать работу таких устройств.

Simtera вчера и сегодня?

Вот уже больше года прошло с того момента как запустили первую версию Simtera. К выходу предыдущей версии мы продемонстрировали [проект симуляции работы операционной системы FX-RTOS](#). Процессор был написан на VHDL и после демонстрации работы симулятора мы сконфигурировали ПЛИС и убедились в том, что работа кода в симуляторе и в реальном устройстве совпадает. На тот момент единственным языком, который «понимала» Simtera был VHDL. В новом же релизе добавлена работа с Verilog, и поддержка мультязычных

VHDL/Verilog проектов. Это не просто расширение возможностей. Мы добились того, что над одним проектом могут совместно работать как Verilog, так и VHDL разработчики.

Мы и наши конкуренты

Основными аналогами Simtera на рынке являются - ModelSim (Mentor Graphics, США) и Active-HDL (Aldec, США). Сводная таблица сравнения первой версии Simtera с ModelSim-Altera 10.1 и Aldec Active-HDL 8.2 Expert Edition приведена [тут](#). Основными причинами использования именно нашего продукта является:

1. Simtera является "золотой серединой" между ModelSim и Active-HDL. Simtera поддерживает VHDL и Verilog спецификации и осуществляет моделирование быстрее чем ModelSim (ModelSim - работает точно, но медленно). В то же время результаты моделирования точнее, чем у Active-HDL, который позволяет быстро получить результаты моделирования, которые, тем не менее, обладают большими погрешностями,
2. Удобство работы в графическом интерфейсе, т.к. Simtera не только симулятор, но и среда разработки. В систему входит текстовый редактор с подсветкой синтаксиса, удобный интерактивный осциллограф. Много внимания уделяется моментам, позволяющим упростить работу HDL - разработчику, и тому, чтобы работать с Simter'ой было удобно и приятно,
3. В стандартном пакете поставки продукта возможно использование совместно и Verilog, и VHDL. В то время как ModelSim в стандартном пакете предоставляет компиляцию проектов, выполненных лишь на одном из языков.

Примеры работы в программном пакете Simtera

??? ?????? ??????? Verilog ? VHDL

Для демонстрации работы в текстовом редакторе и осциллографе Simtera мы приведем несложные проекты,

написанные на Verilog и VHDL. В листинге приведен пример, в котором по положительному фронту тактового сигнала (clk) идет приращение счетчика (cnt) :

Листинг с кодом Verilog и VHDL представлен в спойлере №1.

Листинг №1 (кликните для показа/скрытия)

Verilog

VHDL

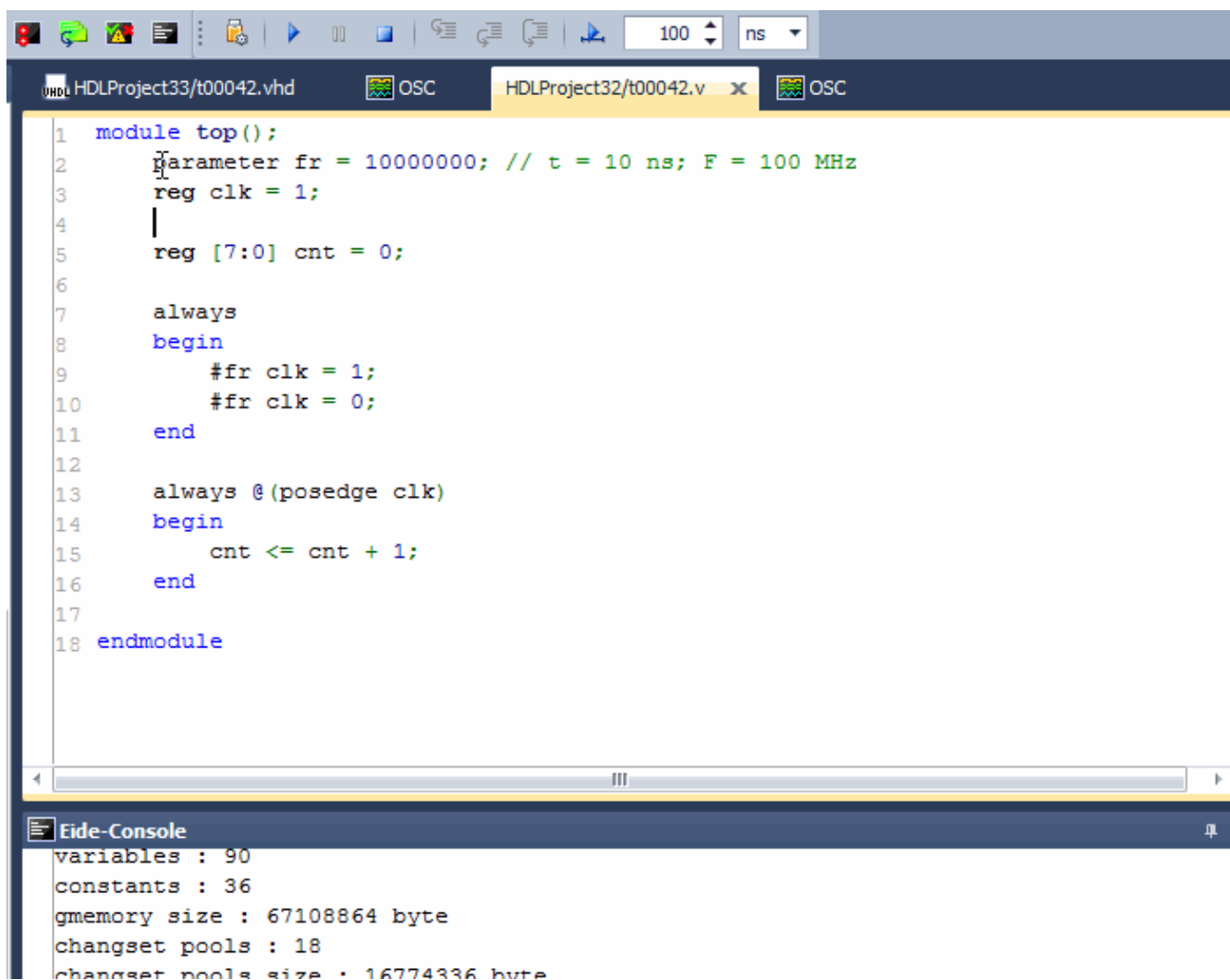
```
module top();
    parameter fr = 100000000ee;
    ; // t = 10 ns; F = 100 MHz
    reg clk = 1; reg [7:0] cnt;
    cnt = 0;
    always # (fr) clk = 1;
    # (fr) clk = 0;
    always @ (posedge clk) cnt = cnt + 1;
endmodule

library ieee;
use ieee.std_logic_1164.all;
use ieee.cnumeric_std.all;
entity bety_top is
    architecture top of bety_top is
        signal clk : std_logic;
        signal cnt : unsigned(7 downto 0);
    begin
        process (clk)
            wait for 10 ns;
            clk <= '0';
            wait for 10 ns;
            clk <= '1';
        end process;
        process (clk)
            begin
                if (rising_edge(clk)) then
                    cnt <= cnt + to_unsigned(1,8);
                end if;
            end process;
    end architecture top;
end entity bety_top;
```

Ниже представлена работа в текстовом редакторе и в

осциллографе Simtera. В баре отображения окон - 4 вкладки. 2 вкладки - тесты программ на Verilog и VHDL (приведены выше), 2 вкладки - осциллограммы к проектам.

Листинг №2 (кликните для показа/скрытия)



The screenshot shows a software interface with a code editor and a console. The code editor displays the following Verilog code:

```
1 module top();
2     parameter fr = 10000000; // t = 10 ns; F = 100 MHz
3     reg clk = 1;
4     |
5     reg [7:0] cnt = 0;
6
7     always
8     begin
9         #fr clk = 1;
10        #fr clk = 0;
11    end
12
13    always @(posedge clk)
14    begin
15        cnt <= cnt + 1;
16    end
17
18 endmodule
```

The IDE console at the bottom shows the following output:

```
Hide-Console
variables : 90
constants : 36
memory size : 67108864 byte
changset pools : 18
changset pools size : 16774336 byte
```

Как видно, в списке отображения данных открыты 4 окна. Слева направо: HDLProject33/t00042.vhd, OSC [осциллограф для VHDL проекта], HDLProject32/t00042.v, OSC[осциллограф для Verilog проекта]. В файлах HDLProject33/t00042.vhd и HDLProject32/t00042.v - VHDL и Verilog код, представленный в листинге (см. Спойлер №1). В осциллографах же можно наблюдать что по переднему фронту сигнала clk идет инкрементация счетчика cnt.

На gif изображении также демонстрируется текстовой редактор (выделение переменных, выделение текста), демонстрируется работа в осциллографе (побитовая развертка шины данных, перевод данных из одной системы отсчета в другую).

В следующей статье мы рассмотрим как в Simtera создать проект, как система выявляет синтаксические и семантические ошибки. Также одну из статей из серии обязательно посвятим мультязычным и многомодульным проектам.

Следите за новостями проекта.

С уважением, Никита Малышев.