Семинар 1.
Моделирование схем на языке Verilog. Симуляция и тестирование схем.

Математические модели и методы синтеза СБИС
Весна 2015

Языки описания аппаратного обеспечения

- Hardware Description Language (HDL) – язык описания структуры и функционирования аппаратного обеспечения цифровой системы на различных уровнях абстракции.
- Основные HDL:
  - Verilog
  - SystemVerilog
  - VHDL
  - SystemC
  - …
- http://en.wikipedia.org/wiki/Hardware_description_language

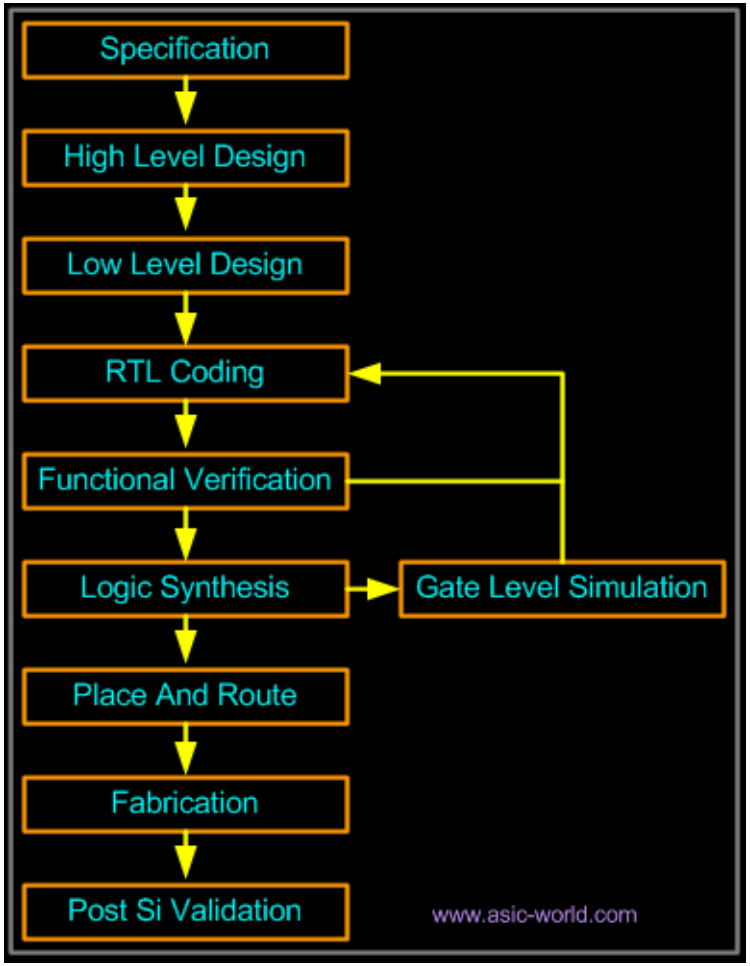
Стили разработки аппаратного обеспечения

- Снизу-вверх
  - Сначала отдельно проектируются базовые элементы, а потом из простых элементов проектируются (собираются) более сложные элементы.
- Сверху-вниз
  - Сначала проектируются системы более высокого уровня, а потом рекурсивно проектируются их компоненты более низкого уровня.

Основные уровни абстракции аппаратного обеспечения

- Поведенческий уровень (Behavioral Level)
  - Описание устройства при помощи набора последовательных алгоритмов, работающих одновременно и согласованно.
- Уровень регистровых передач (Register-Transfer Level)
  - Описание устройства при помощи заданного набора операций преобразования значений, хранящихся в регистрах. По сути устройство описывается в виде иерархии взаимосвязанных автоматов.
- Схемный (функциональный) уровень (Gate Level)
  - Описание устройства при помощи схемы из функциональных элементов в заданной библиотеке элементов.

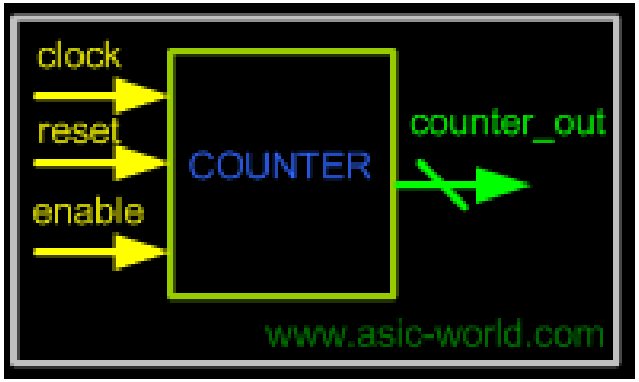
Упрощенный маршрут проектирования



Язык Verilog – Hello world!

```
1  module hello_world;
2    initial begin
3      $display ("Hello world!");
4      #10 $finish;
5    end
6  endmodule
```

Пример моделирования на языке Verilog – 4-х битовый счетчик



- Входы
  - Clock – вход тактового генератора
  - Reset – сброс значения счетчика (сброс при «1»)
  - Enable – включение счетчика (при «1» счетчик работает)
- Выходы
  - Counter_out – значение счетчика
- Функция
  - Счетчик увеличивается на 1-цу за каждый «такт» тактового генератора

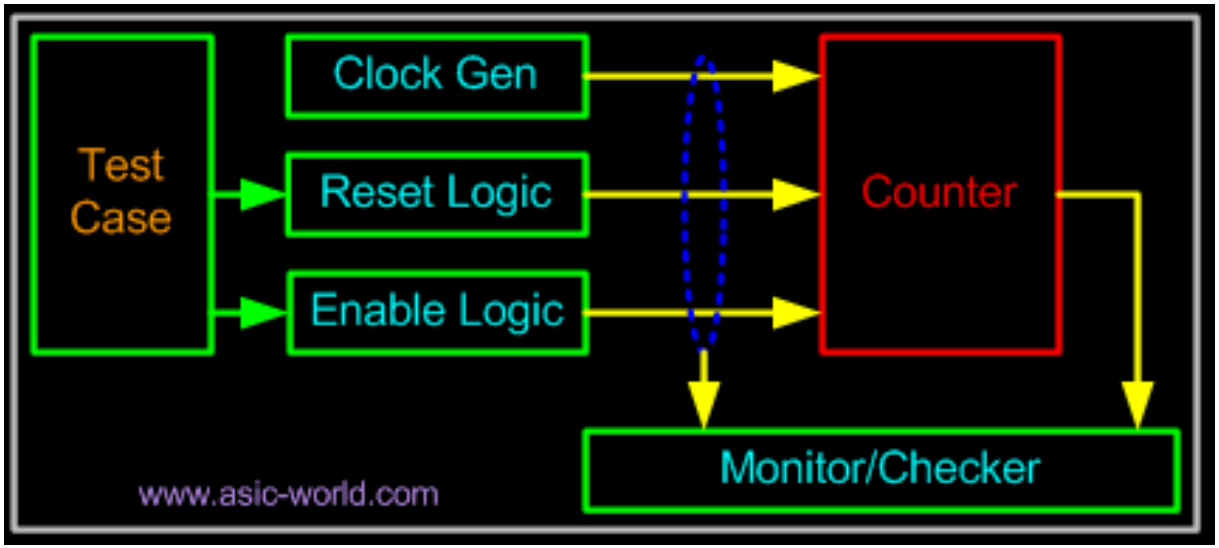
Пример моделирования на языке Verilog – 4-х битовый счетчик

```
1  module counter (clock, reset, enable, counter_out);
2
3    input clock;
4    input reset;
5    input enable;
6
7    output [3:0] counter_out;
8
9    wire clock;
10   wire reset;
11   wire enable;
12
13   reg [3:0] counter_out;
14
```

Пример моделирования на языке Verilog – 4-х битовый счетчик

```
15    always @ (posedge clock)
16      begin: COUNTER
17        if (reset == 1'b1) begin
18          counter_out <= #1 4'b0000;
19        end
20        else if (enable == 1'b1) begin
21          counter_out <= #1 counter_out + 1;
22        end
23      end
24  endmodule
```

Пример моделирования на языке Verilog – тестирование



Пример моделирования на языке Verilog – тестирование

```
1  `include "counter.sv"
2
3  module counter_tb();
4    reg clock, reset, enable;
5    wire [3:0] counter_out;
```

Пример моделирования на языке Verilog – тестирование

```
7   initial begin
8     $dumpfile("dump.vcd");
9     $dumpvars(1);
10
11    $display ("time\t clk\t reset\t enable\t counter");
12    $monitor ("%g\t %b\t %b\t %b\t %b",
13              $time,
14              clock,
15              reset,
16              enable,
17              counter_out
18              );
19
```

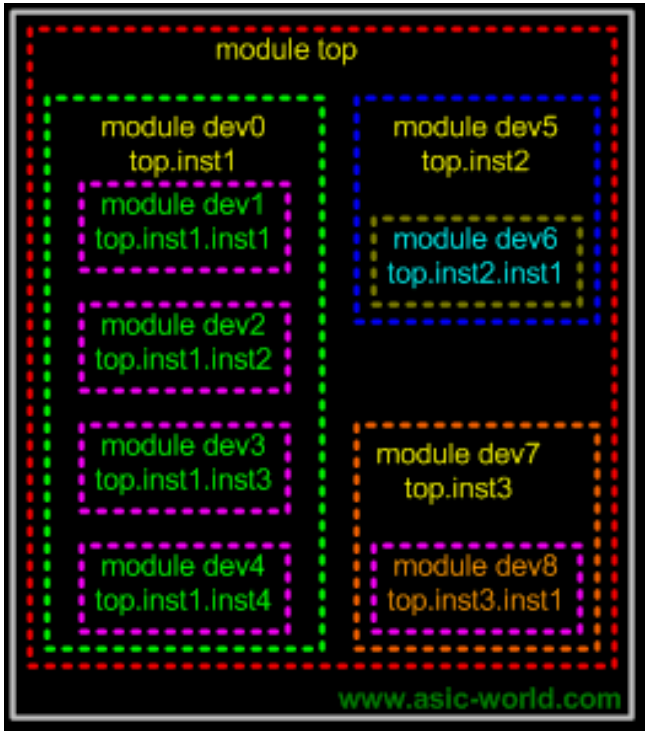
Пример моделирования на языке Verilog – тестирование

```
20      clock = 1;
21      reset = 0;
22      enable = 0;
23      #5 reset = 1;
24      #10 reset = 0;
25      #10 enable = 1;
26      #100 enable = 0;
27      #5 $finish;
28    end
29
```

Пример моделирования на языке Verilog – тестирование

```
30    always begin
31      #5 clock =~ clock;
32    end
33
34    counter test_counter (
35      clock,
36      reset,
37      enable,
38      counter_out
39    );
40
41  endmodule
```

Язык Verilog - модули



- Основные строительные блоки аппаратного описания на языке Verilog
- Описание представляет собой вложенную иерархию модулей
- Инстанцирование модулей

Язык Verilog – объявление и инстанцирование модулей

```
1  module device(device_output, input_1, input_2);
2    ...
3  endmodule
4
5  module another_device();
6    wire wire_1, wire_2;
7    reg register;
8    ...
9    device device_insance(register, wire_1, wire_2);
10   ...
11 endmodule
```

Язык Verilog - сети

- Сети позволяют хранить и передавать сигналы между модулями.
- Основные типы сетей:
  - Провода (wire)
  - Регистры (reg)
- Провода могут иметь ветвление
- Специальные типы сетей: tri, wor, trior, wand, triand, tri0, tri1, supply0, supply1, trireg.

Язык Verilog – сети

```
1  module counter (clock, reset, enable, counter_out);
2
3    input clock;
4    input reset;
5    input enable;
6
7    output [3:0] counter_out;
8
9    wire clock;
10   wire reset;
11   wire enable;
12
13   reg [3:0] counter_out;
14
```

Язык Verilog - порты

- Порты задают входы и выходы модулей
  - input, output, inout
- Связывают модули с сетями
- Все модули имеют порты (кроме модулей верхнего уровня иерархии)
- Явная (by name) и неявная (by order) «привязка» портов

Язык Verilog – объявление портов

```
1  module bit_adder(in_1, in_2, carry_in, out, carry_out);
2    ...
3  endmodule
4
5  module adder(a, b, c);
6    input [1:0] a, b;
7    output [2:0] c;
8
9    wire [1:0] a, b;
10   wire [2:0] c;
11   wire t_0, t_1;
12   ...
13   assign t_0 = 0;
14   bit_adder add_0 (
15     .in_1 (a[0]),
16     .in_2 (b[0]),
17     .carry_in (t_0),
18     .out (c[0]),
19     .carry_out (t_1)
20   );
21
22   bit_adder add_1 (a[1], b[1], t_1, c[1],c[2]);
23   ...
24 endmodule
```