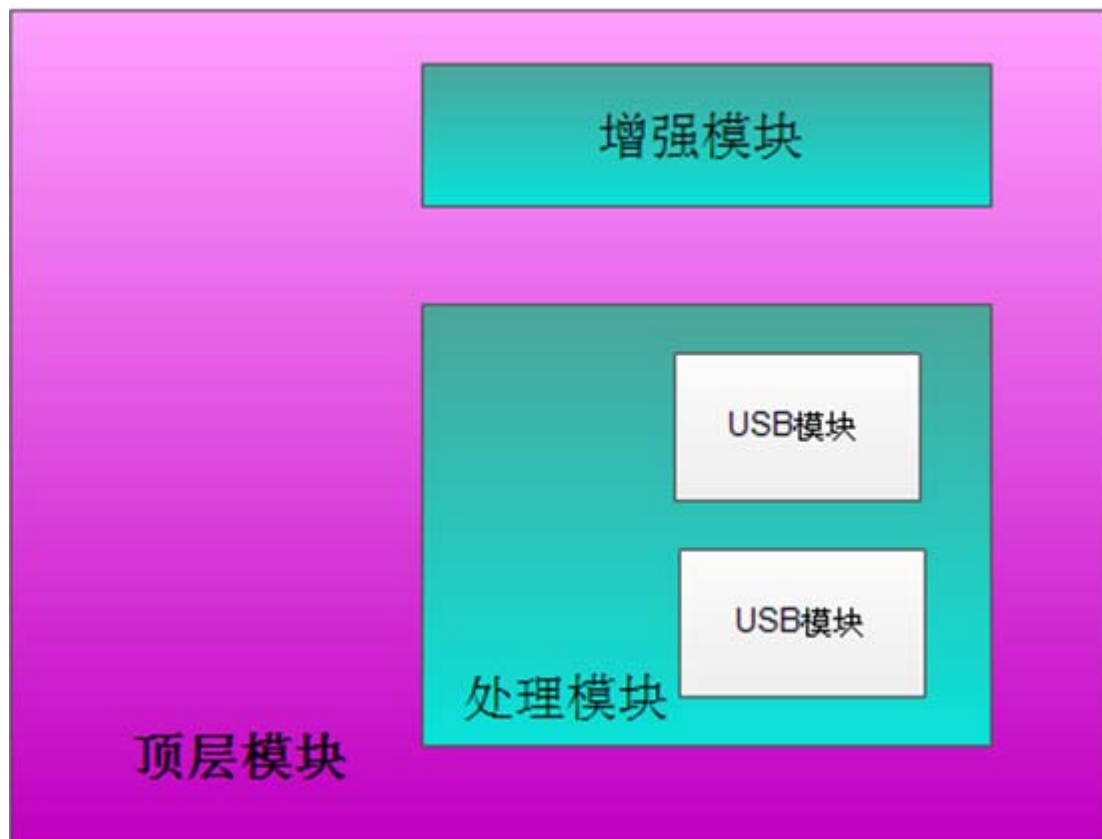


verilog 快速掌握之模块例化

一、什么是例化

1) FPGA 逻辑设计中通常是一个大的模块中包含了一个或多个功能子模块, **verilog** 通过模块调用或称为模块实例化的方式来实现这些子模块与高层模块的连接, 有利于简化每一个模块的代码, 易于维护和修改。

2) 如下图, 当有了例化的方法之后, 当模块中需要两个同样功能的 **USB** 模块时, 我们只需要编写好一个 **usb** 模块的代码, 之后通过例化, 就可以将俩个 **usb** 模块添加到顶层模块之中。(注意, 我们要有硬件思维, 例化一个模块, 那么就有一个硬件与之对应。例化多个, 就有多个硬件。)

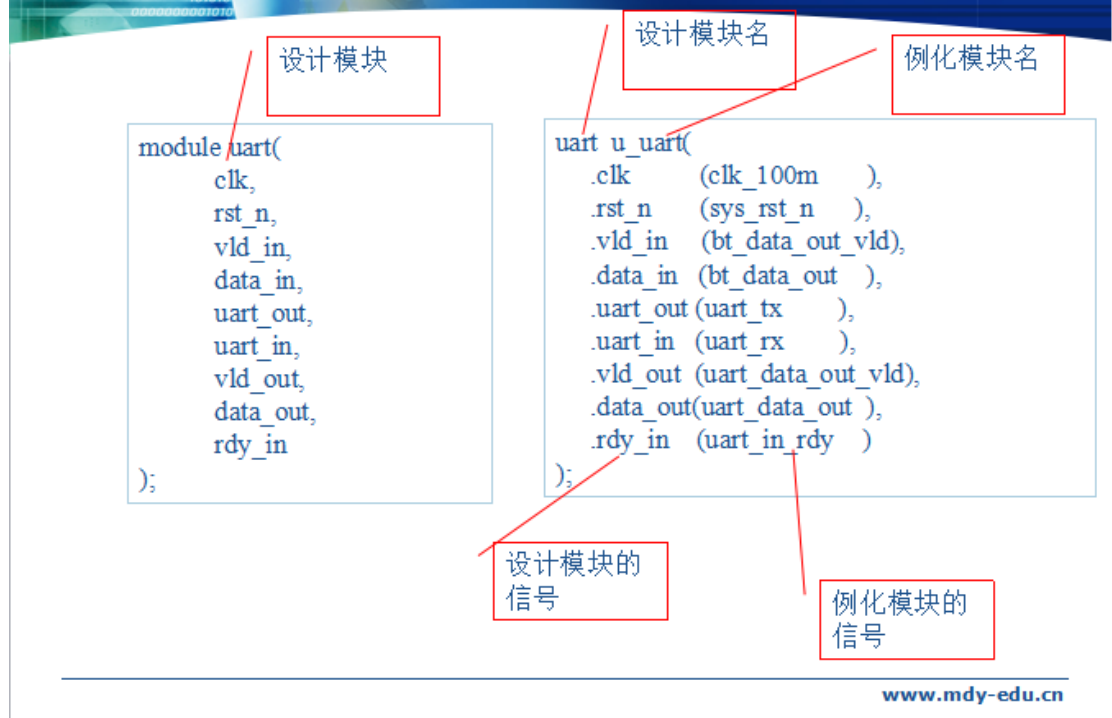


二、例化方法

1) 首先我们设计好我们需要的功能模块, 如下图左边的 **uart** 模块;

2) 之后我们在大模块例化 **uart** 模块, 如下图右边的例化格式, 信号列表中 “.” 之后的信号是 **uart** 模块定义的信号, 括号内的信号则是在大模块中声明的信号, 这样就将大模块的信号与 **uart** 模块信号一一对应起来。

三、例化方法



三、参数例化

1) 模块例化还有一个好处就是在调用的时候还可以传递/改变参数。如下图，我们设计了一个 `uart` 模块，他的数据位宽只有 8 位，但是如果我们需要一个 16 位宽的 `uart` 模块，这时候我们可以通过参数例化来减少重新编写模块的工作量。

2) 我们定义数据位宽为 `DATA_W=8`

```

module uart(
    clk,
    rst_n,
    vld_in,
    data_in,
    uart_out,
    uart_in,
    vld_out,
    data_out,
    rdy_in
);

parameter DATA_W = 8;
  
```

3) 在例化的时候我们可以如下图，在模块名后面加上 “# (.DATA_W(16))”，就可以将数据位宽定义为 16 位。

```
uart#(.DATA_W(16)) u_uart(  
    .clk      (clk_100m      ),  
    .rst_n    (sys_rst_n     ),  
    .vld_in   (bt_data_out_vld),  
    .data_in  (bt_data_out   ),  
    .uart_out (uart_tx        ),  
    .uart_in  (uart_rx        ),  
    .vld_out  (uart_data_out_vld),  
    .data_out (uart_data_out  ),  
    .rdy_in   (uart_in_rdy   )  
);
```

注意，大家务必掌握参数传递的方法，点拨 FPGA 课程后面有一部分练习要计时到 1 秒时间，如果 modelsim 真的要仿真 1 秒，跑一个用例就要花费好几分钟或者几个小时，这样的低效率是难以忍受的。我们仿真时，在测试文件改变 1 秒的参数值，将它变得很小，这样就能加快仿真迅速。详细情况，学到相应章节自然会清楚的。