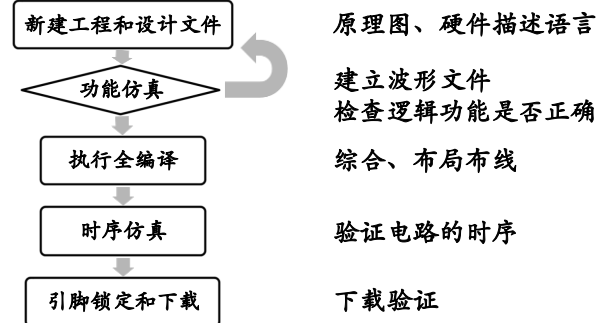


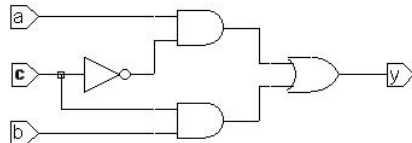
## EDA讲座2

- 时序仿真和层次化设计
- 硬件描述语言2
- EDA作业二

### 设计流程



2



a	b	c	y
X	X	0	a
X	X	1	b

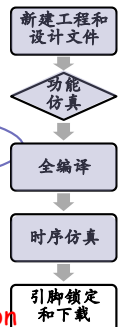
3

### 1. 时序仿真

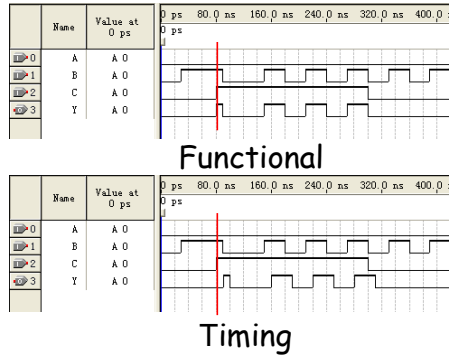
在全编译之后验证设计的时序性能

- ✓ 建立波形文件
- ✓ 添加待观测信号节点
- ✓ 添加输入激励
- 打开波形文件\*.VWF，修改激励
- 在 **Simulation → Options** 中指定仿真工具  
选择 **QuartusII Simulator**
- 执行时序仿真 **Simulation → Run timing Simulation**
- 观察仿真结果

Edit → Set End Time  
10ns ~ 100us



4



5

## 2.封装元件和层次化设计

- 封装成同名的元件符号

File→Creat/Update →Creat Symbol Files For Current File

- 新建原理图，添加封装好的元件mux21

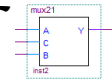
- 绘图后，将原理图命名为mux4\_21并保存

- 将其设定为顶层文件：

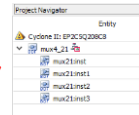
导航栏Files页签：选中mux4\_21.bdf，

右键菜单Set as Top-level Entity

导航栏Hierarchy页签：顶层文件即改为mux4\_21

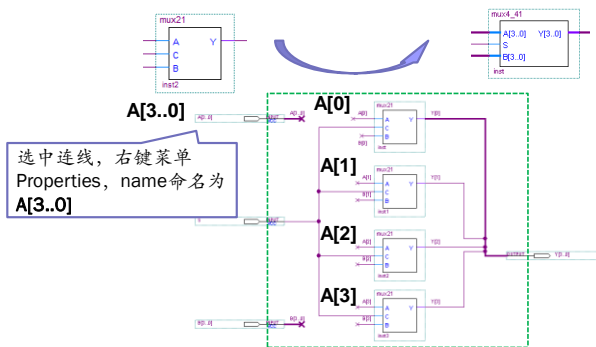


编译后导航栏中的层次：



6

### 扩展：原理图



7

### 扩展：硬件描述语言

```
module mux21 (a,b,c,y);
input a,b,c;
output y;

reg y;

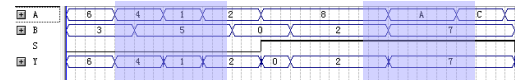
always @ (a,b,c)
begin
if (c==0) y<=a;
else
y<=b;
end
endmodule
```



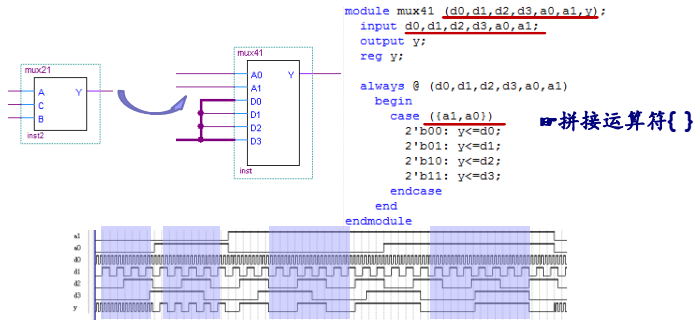
```
module mux4_21 (A,B,S,Y);
input [3:0] A;
input [3:0] B;
input S;
output [3:0] Y;

reg [3:0] Y;

always @ (A,B,S)
begin
if (S==0) Y<=A;
else
Y<=B;
end
endmodule
```



## 扩展：硬件描述语言



## 3.硬件描述语言2

### 一、组合电路

1. 电路模块的基本结构
2. 电路模块的描述方式

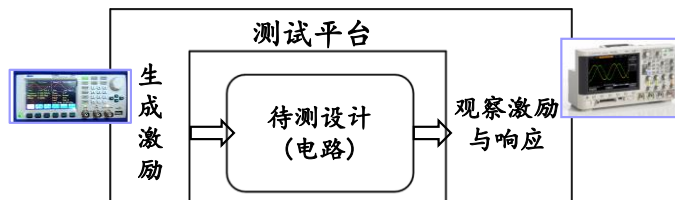
### 二、测试平台

### 三、时序电路

1. 同步复位和异步复位的D触发器
2. 状态机

## 测试平台 (Testbench)

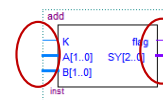
- 为系统设计提供激励的HDL描述
- 测试和观察在此激励下的响应



## 电路设计vs.测试文件

测试文件add\_tb.v

电路设计add.bdf/add.v



```

`timescale 10ns/1ns
module add_tb;
  wire [2:0]so;
  wire F;
  reg [1:0]a,b;
  reg flag;

  add u1(.A(a),.B(b),.K(k),.SY(so),.flag(F));

  initial
  begin
    a=0;b=0;k=0;
    #20 a=0;b=1;k=0;
    #20 a=0;b=10;k=0;
    #20 a=0;b=11;k=0;
    #20 a=01;b=00;k=0;
    #20 a=01;b=01;k=0;
    #20 a=01;b=10;k=0;
    #20 a=10;b=11;k=1;
    #20 a=11;b=00;k=1;
    #20 a=11;b=01;k=1;
    #20 a=11;b=10;k=1;
    #20 a=11;b=11;k=1;
  end
  initial
  begin
    $monitor ("k=%d a=%d b=%d so=%d F=%d",k,a,b,so,F);
  end
endmodule

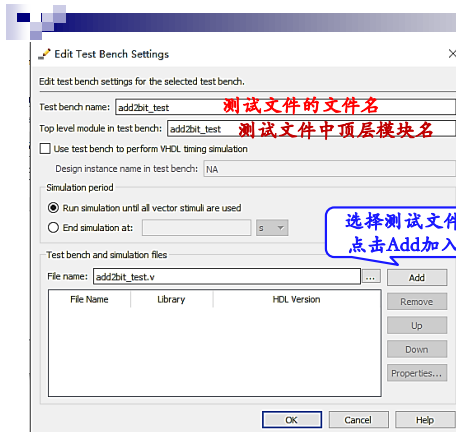
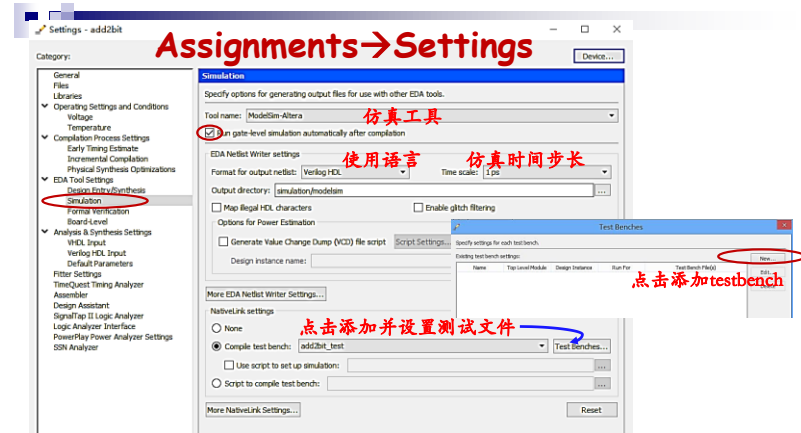
```





## 关联仿真

- 仿真设置, **Assignments→Settings**
  - 仿真工具
  - 语言及仿真步长
  - 设置并添加测试文件

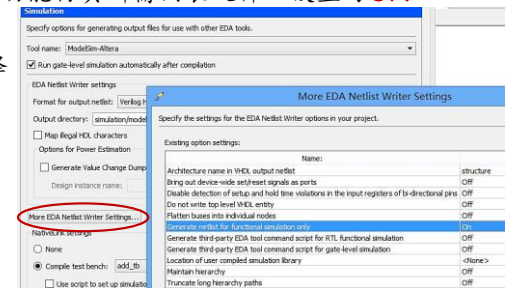


测试文件的文件名  
测试文件add2bit\_test.v  
测试文件中顶层模块名

```
timescale 10ns / 1ns
module add2bit_test;
```

## 执行功能仿真

- **Settings**页中, 点击**More EDA Netlist Writer Settings...**“生成功能仿真所需网表文件”设置为**ON**
- 设置完成后全编译并启动 modelsim



## 执行时序仿真

- Settings页中，点击More EDA Netlist Writer Settings... “生成功能仿真所需网表文件” 设置为OFF
- 设置完成后全编译并启动modelsim

## \*基于电路设计生成测试文件

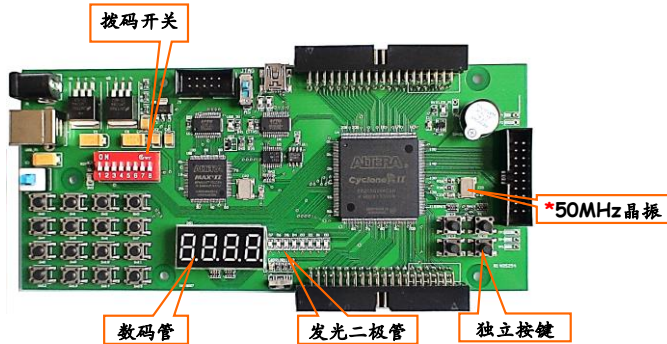
- 电路设计和测试文件，二者名字不能相同，如：add.v和testbench.v
- 使用模板生成测试文件\*.vt，步骤如下：
  - ☞ 执行Processing→Start→Start TestBench Template Writer，生成的模板文件存放路径：project文件夹中/simulation/modelsim/\*.vt
  - ☞ 修改\*.vt文件名，必须与module名相同，如add\_vlg\_tst
  - ☞ 打开vt文件，添加激励

## ■ 可利用的资源：

- ☞ 教学视频（课程文件--EDA视频）
- ☞ 文档资料（课程文件-- EDA文档）
- ☞ AI



EDA作业二

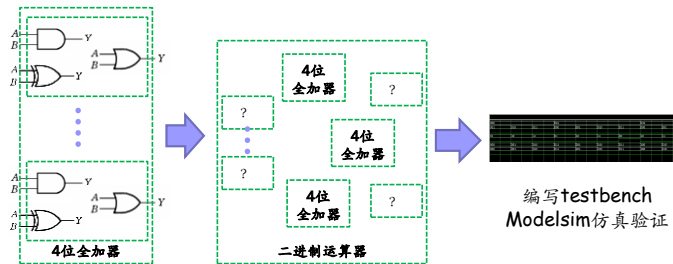


## 实验任务:

- 设计一个运算电路，实现  $S=M+N$ 。M、N为3位有符号数。  
运算数输入：拨码开关、按键  
运算结果显示：数码管、发光二极管
- 任务分解：
  - ✓ 二进制运算器
  - ✓ 4位数码管驱动电路

## 任务1：二进制运算器（原理图）

自下而上的模块化、层次化的设计

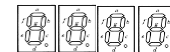


## 任务2：4位数码管的驱动电路

$$S=M+N$$

DIP1 DIP2	数码管3	数码管2	数码管1	数码管0
00	M	不亮	不亮	不亮
01	不亮	N	不亮	不亮
10	不亮	不亮	S(正负标志)	不亮
11	不亮	不亮	不亮	S(运算结果)

二进制  
运算器

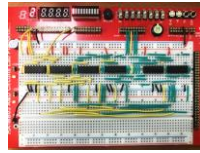




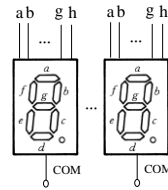
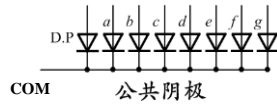
## ■ 数码管静态显示方式

共阴极数码管，高电平有效

段选线、位选线由FPGA引脚直接驱动



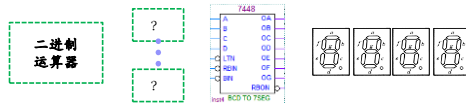
☞ 显示位数较多时，占用I/O



## 任务：4位数码管的驱动电路

$$S=M+N$$

DIP1 DIP2	数码管3	数码管2	数码管1	数码管0
00	M	不亮	不亮	不亮
01	不亮	N	不亮	不亮
10	不亮	不亮	S(正负标志)	不亮
11	不亮	不亮	不亮	S(运算结果)



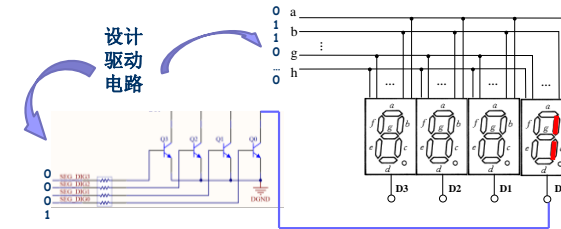
## ■ 数码管动态显示方式

段选线并联在一起，共阴极为选通端

驱动电路端轮流加以高电平，点亮相应的数码管

采用动态扫描的方式，使多位数码管“同时”被点亮

选做



## ■ 任务分解→验收步骤：

☞ 第10周：二进制运算器及其Modelsim仿真

☞ 第11周：数码管显示电路、整体下载

☞ 预先设计，课上答疑、调试、验收为主。

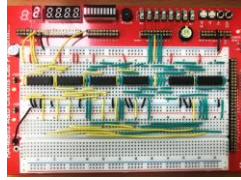
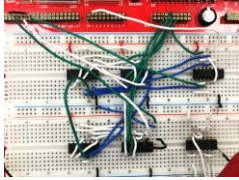
## ■ 提供资源：

☞ 验收细则

☞ 第10周课上讲解验收细则

☞ 答疑

## ■ 第9周实验预告：组合电路（2位全加器）



### 识别和检测器件 → 整体布局 → 合理布线

- ☞ 面包板结构
- ☞ 芯片及其数据手册
- ☞ 检测元件、准备导线

- ☞ 考虑信号传递方向
- ☞ 考虑外设方向
- ☞ 元件的安装便于检查
- ☞ 测量和更换
- ☞ 输入输出端放置标记

- ☞ 逻辑图→连线图→元件引脚编号
- ☞ 布线短而简洁，横平竖直
- ☞ 尽量贴近面包板
- ☞ 布线便于检查、测量

