

数字逻辑设计

实验6 十六进制计算器设计

马世禹



HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ

实验目的

- (1) 掌握自顶向下(Top-Down)的结构化设计方法;
- (2) 培养和锻炼对复杂问题的分析与求解能力;
- (3) 培养和锻炼复杂数字系统的设计能力。



实验内容

设计一个十六进制的计算器，支持加、减、乘、求商，求余，平方六种运算，且支持连续运算功能；

详细要求如下：

A.输入时钟为100MHz，端口为Y18；

B.使用按键开关S1作为异步复位信号，且当S1为1时，计算器将被复位；

C.按键开关S2作为每次计算启动信号（类似等号功能）；

D.操作数由拨码开关SW15-SW0输入，操作数1为SW15-SW8，操作数2为SW7-SW0；

E.连续运算时，操作数2为输入的数据；

F.当前运算功能由拨码开关SW23-SW21来决定：

G.计算结果实时输出到数码管上；

SW23-SW21	Function
3'b000	加
3'b001	减
3'b010	乘
3'b011	求商
3'b100	求余
3'b101	平方



实验原理

➤ 自顶向下的结构化设计方法

自顶向下的设计方法是指在设计过程中，设计开发人员首先从整体上规划整个系统的功能和性能，然后根据具体需求和系统功能对系统进行子模块划分，从而将整个系统分解为规模较小、功能较简单的子模块，并在这个过程中确立子模块之间的交互关系和接口。这个过程可以不断地进行下去，直到最终得到的子模块可以很容易地映射到物理层面实现。

将自顶向下的结构化设计方法应用到数字系统的设计上。首先将一个比较复杂的数字电路划分为多个组成模块，再分别对每个模块建模，然后将这些模块组成一个总模块，完成所需的功能。



实验原理

➤ 自顶向下的结构化设计方法

结构化设计的基本单位是模块，每个模块的组成可以遵循以下规则：

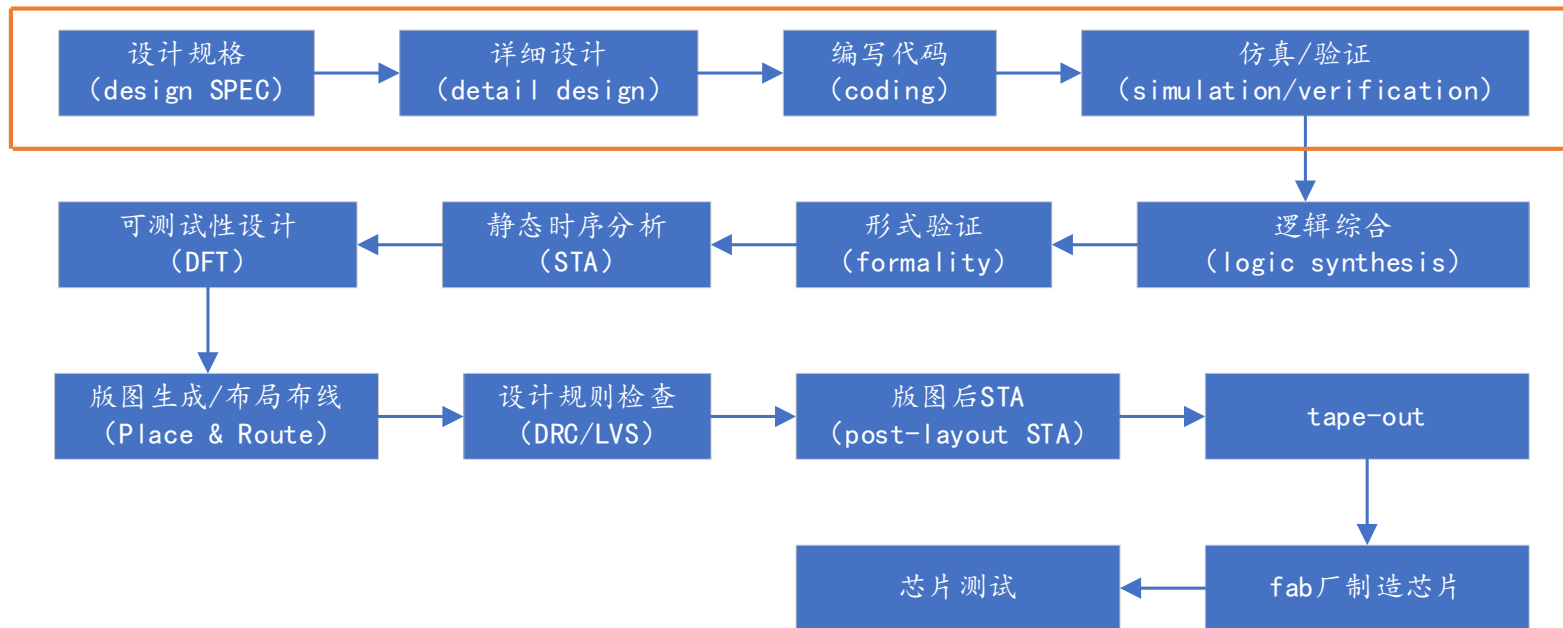
- (1) 在设计中至少被使用两次的逻辑，该逻辑采用实例化方式重用，为增加代码的易读性和整洁性，可将其封装成模块，如译码器、多路选择器等；
- (2) 功能规格十分明确，且与外界的交互信号数目不是很多的，封装成模块，如消抖模块等；
- (3) 现有的一个模块达到了数千行代码的规模，将其拆分成若干和小模块；
- (4) 一个文件中只包含一个模块，便于后期代码维护。



实验原理

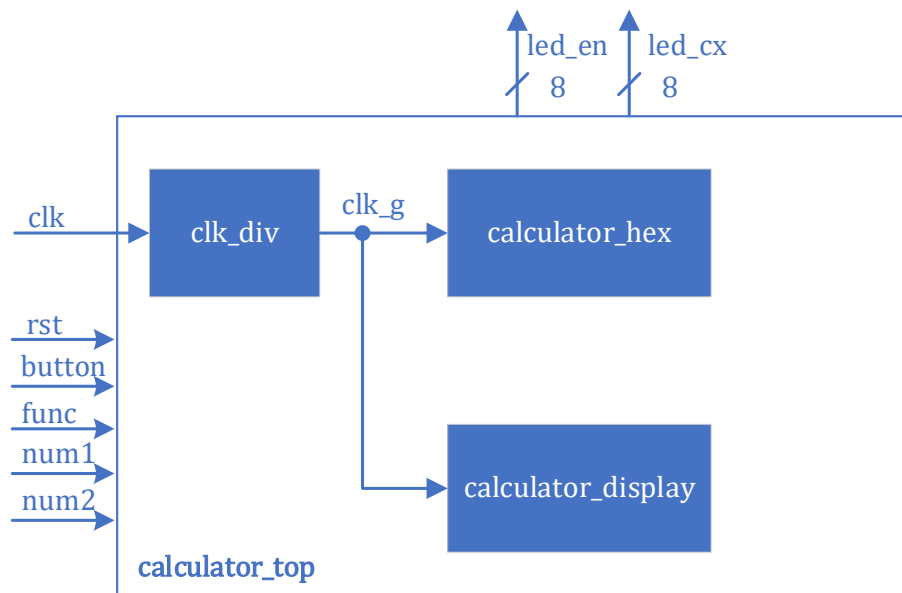
➤ 数字集成电路开发流程

FPGA开发流程



实验原理

➤ 模块划分



可以将整个系统分为3个模块：

clk_div: 用于时钟分频（必须），
分频频率为**10MHz**；

calculator_hex: 实现计算器计算
功能；

calculator_display: 实现计算器结
果在数码管上显示；

实验原理

► 接口定义

Name	I/O	Width	Description
clk	input	1	时钟信号 (100MHz)
rst	input	1	复位信号
button	input	1	计算启动信号
func	input	3	计算功能选择信号
num1	input	8	操作数 1
num2	input	8	操作数 2
led_en	output	8	数码管显示控制信号
led_ca	output	1	数码管显示控制信号
led_cb	output	1	数码管显示控制信号
led_cc	output	1	数码管显示控制信号
led_cd	output	1	数码管显示控制信号
led_ce	output	1	数码管显示控制信号
led_cf	output	1	数码管显示控制信号
led_cg	output	1	数码管显示控制信号
led_dp	output	1	数码管显示控制信号



仿真需求

- calculator_top文件中, 例化的calculator_hex的cal_result[31:0]接口必须保留, 该端口表示每一次计算结果的输出;

```
calculator_hex u_calculator_hex (  
    .clk      (clk_g      ),  
    .rst      (rst        ),  
    //add your own code  
    .cal_result (cal_result)  
);
```



实验步骤

- ❑ 创建工程，工程名为calculator_hex;

- ❑ 编写并添加设计文件

calculator_hex.v/calculator_display.v/calculator_top.v;

- ❑ 创建时钟IP，clk_div，集成到calculator_top.v中；

- ❑ 添加提供的仿真文件testbench.v，并完成仿真；

- ❑ 编写并添加约束文件，并综合实现，生成比特流；

- ❑ 将生成的比特流下载到开发板验证；



验收要求

- ☐ 计算器仿真通过（2分）
- ☐ 计算器开发板实现（2分）
- ☐ 实验报告（2分）



提交要求

- ❑ 提交时间：详见网页指导书
- ❑ 提交格式：学号_姓名.zip
- ❑ 注意：如有出现雷同，雷同者均不得分！



开始实验



HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ