

# 中山大学数据科学与计算机学院本科生实验报告

## (2018 学年第二学期)

课程名称：数字电路与逻辑设计实验

任课教师：郭雪梅

助教：林鸿鑫

年级&班级	18 级计科 7 班	专业(方向)	计算机类
学号	18340181	姓名	谢俊杰
电话	16607657742	Email	xiej8@mail2.sysu.edu.cn
开始日期	2019/5/18	完成日期	2019/5/18

### 一、实验题目

基于 vivado 设计 8 位的加减法器

### 二、实验目的

- 熟悉 Vivado IP 核的功能与使用方法。
- 掌握用 MSI 设计的组合逻辑电路的方法。

### 三、实验内容

#### 1. 实验步骤

生成与门逻辑：

- (1) 新建工程文件，添加8位的加减法器的源文件，代码如下：

Σ Project Summary × addsub.v \*

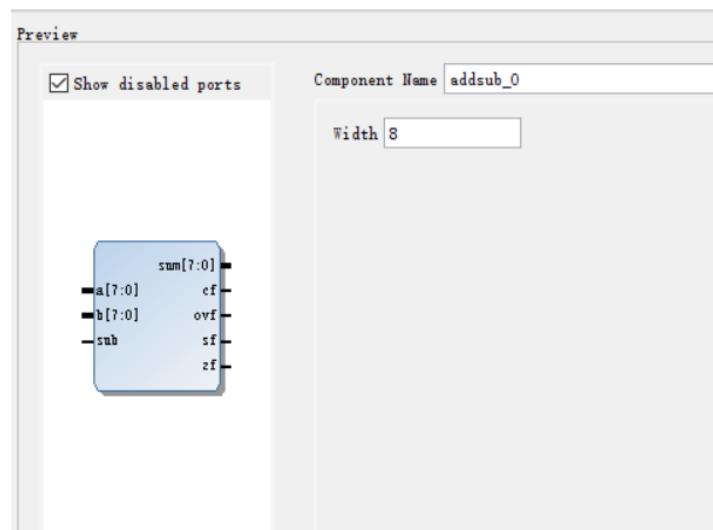
D:/code/freshman/digital/XUP\_LIB/addsub/addsub.v

```

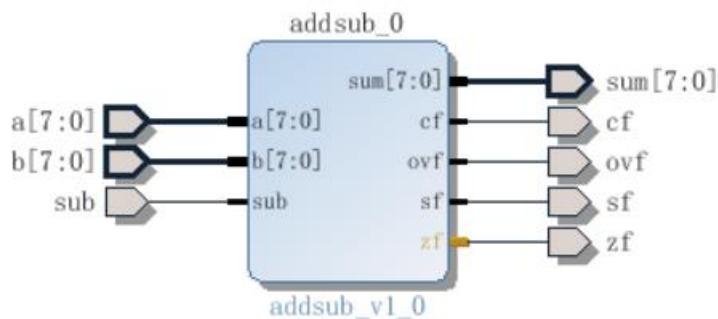
1 `timescale 1ns / 1ps
2
3 module addsub
4 #(parameter WIDTH=8)           //指定数据宽度参数，缺省值是8
5 (
6     input [(WIDTH-1):0] a,      // 缺省位数由参数WIDTH决定
7     input [(WIDTH-1):0] b,
8     input sub,                // =1为减法
// 9     output [(WIDTH-1):0] sum, // 进位标志
10    output cf,                // 溢出标志
11    output ovf,               // 符号标志
12    output sf,                // 为0标志
13    output zf                 // 加法标志
14 );
15 // 添加自己的代码
16 wire [(WIDTH-1):0] subb, subb1;
17     wire cf2; // 进位
18     assign subb1 = b ^ {WIDTH{sub}}; // 对于减法是取反
19     assign subb = subb1 + sub; // 对于减法是加1, sub=1 (减法) sub=0 (加法)
20
21     assign {cf2, sum} = a + subb;
22     assign sf = sum[WIDTH-1];
23     assign zf = (sum == 0) ? 1 : 0 ;
24     assign cf = cf2 ^ sub;
25     assign ovf = (a[WIDTH-1] ^ sum[WIDTH-1]) & (subb[WIDTH-1] ^ sum[WIDTH-1]);
26
27
28 endmodule
29

```

(2) 封装制定IP核，在package IP中添加artix7支持的元器件，所封装的IP核端口如图



(3) 再次重新创建调用IP核的工程文件，并在block design中添加相应端口：



(4) 生成相应的顶层文件addsub8\_wrapper.v，内容如下：

```
Diagram X addsub8_wrapper.v X
D:/code/freshman/digital/addsub8/addsub8.srs/sources_1/bd/addsub8/hdl/addsub8_wrapper.v
1 //Copyright 1986-2015 Xilinx, Inc. All Rights Reserved.
2 //
3 //Tool Version: Vivado v.2015.4 (win64) Build 1412921 Wed Nov 18 09:43:45 MST 2015
4 //Date      : Sat May 18 17:08:10 2019
5 //Host      : LAPTOP-E77AAK9N running 64-bit major release (build 9200)
6 //Command   : generate_target addsub8_wrapper.bd
7 //Design    : addsub8_wrapper
8 //Purpose   : IP block netlist
9 //
10 `timescale 1 ps / 1 ps
11
12 module addsub8_wrapper
13   (a,
14   b,
15   cf,
16   ovf,
17   sf,
18   sub,
19   sum,
20   zf);
21   input [7:0]a;
22   input [7:0]b;
23   output cf;
24   output ovf;
25   output sf;
26   input sub;
27   output [7:0]sum;
28   output zf;
29
30   wire [7:0]a;
31   wire [7:0]b;
```

```
32  wire cf;
33  wire ovf;
34  wire sf;
35  wire sub;
36  wire [7:0]sum;
37  wire zf;
38
39 addsub8 addsub8_i
40     (.a(a),
41      .b(b),
42      .cf(cf),
43      .ovf(ovf),
44      .sf(sf),
45      .sub(sub),
46      .sum(sum),
47      .zf(zf));
48 endmodule
49
```

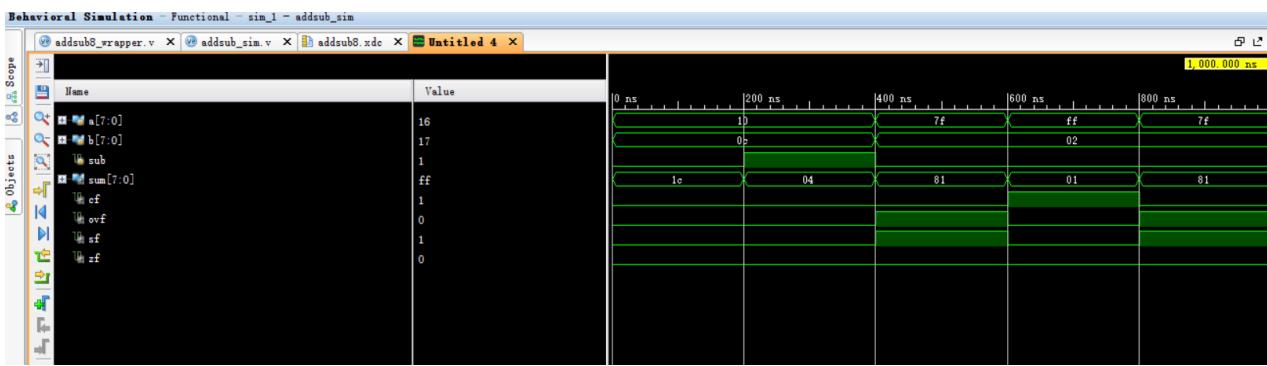
(5) 添加相应的仿真文件，进行功能逻辑仿真检测，代码如下：

The screenshot shows a software interface for Verilog simulation. The title bar says "addsub\_sim.v". The main area displays the Verilog code:

```
5   reg [7:0] b = 7'd12;
6   reg sub = 0;
7
8   //output
9   wire [7:0] sum;
10  wire cf;
11  wire ovf;
12  wire sf;
13 //  wire zf;
14 addsub8_wrapper addsub8
15     (.a(a),
16      .b(b),
17      .cf(cf),
18      .ovf(ovf),
19      .sf(sf),
20      .sub(sub),
21      .sum(sum),
22      .zf(zf));
23 // initial
24 addsub #(8) U (a, b, sub, sum, cf, ovf, sf, zf);
25 initial begin
26 #200 sub = 1;
27 #200 begin a = 8'h7f; b = 8'h2; sub = 0; end
28 #200 begin a = 8'hff; b = 8'h2; sub = 0; end
29 #200 begin a = 8'h7f; b = 8'h2; sub = 0; end
30 #200 begin a = 8'h16; b = 8'h17; sub = 1; end
31 #200 begin a = 8'hff; b = 8'h1; sub = 0; end
32 #200 begin a = 8'hff; b = 8'h1; sub = 0; end
33 end
34 endmodule
35
```

## 四、实验结果

仿真结果如图：



根据对仿真所加条件和结果的分析且由于仿真条件加减法均涉及，可知所设计的8位的加减法器符合其逻辑功能。

## 五、实验感想

在该次实验后，对于 Vivado 的使用也更加熟悉了，也学习到对于仿真文件的设计，既要保证所加条件的全面性，又要顾及其可行性，同时需要注意仿真文件中对于相应文件名的命名书写和要顾及部分对 IP 核的对应关系，且非常容易出错，否则，仿真不能顺利进行。同时对与 Verilog 的语法也略有了解。但是对于源文件代码的设计的能力还是有很大的欠缺。