

# 组成原理课程第一次实报告

## 实验名称：加法器及减法器

学号：22122266 姓名：张恒硕 班次：0416

### 一、实验目的

1. 熟悉 LS-CPU-EXB-002 实验箱和软件平台。
2. 掌握利用该实验箱各项功能开发组成原理和体系结构实验的方法。
3. 理解并掌握加法器、减法器的原理和设计。
4. 熟悉并运用 verilog 语言进行电路设计。
5. 为后续设计 cpu 的实验打下基础。

### 二、实验内容说明

1. 阅读 LS-CPU-EXB-002 实验箱相关文档，了解并熟悉硬件平台,掌握利用显示屏观察特定信号的方法并学习软件平台和设计流程。
2. 熟悉计算机中加法器的原理。
3. 在 verilog 中直接使用 “+” 搭建加法模块，编写相应代码。
4. 仿真编写的代码，得到正确的波形图。
5. 将以上设计作为一个单独的模块，设计一个外围模块去调用之。外围模块中需调用封装好的触摸屏模块，显示两个加数和加法结果，且需要利用触摸功能输入两个加数。
6. 将编写的代码进行综合布局布线，并下载到实验箱中的 FPGA 板上进行演示。
7. 重复上述步骤，通过修改代码，实现减法器，并仿真、演示。

### 三、实验原理图

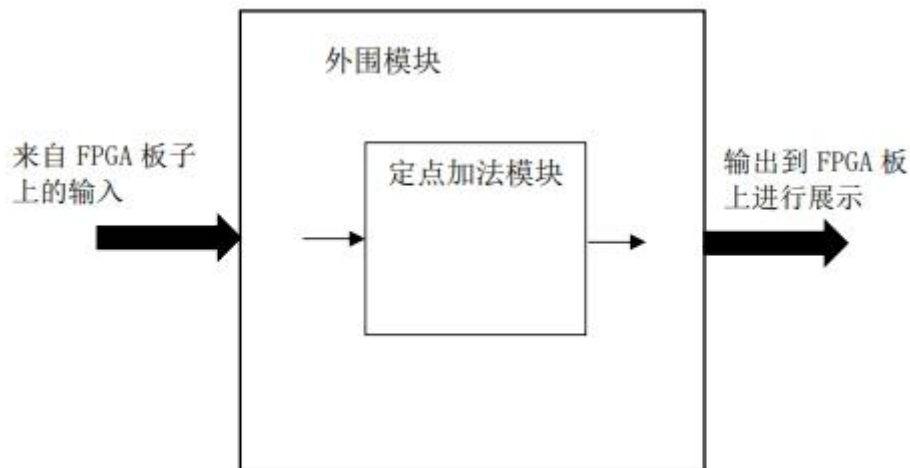


图 2.1 定点加法设计实验的顶层模块大致框图

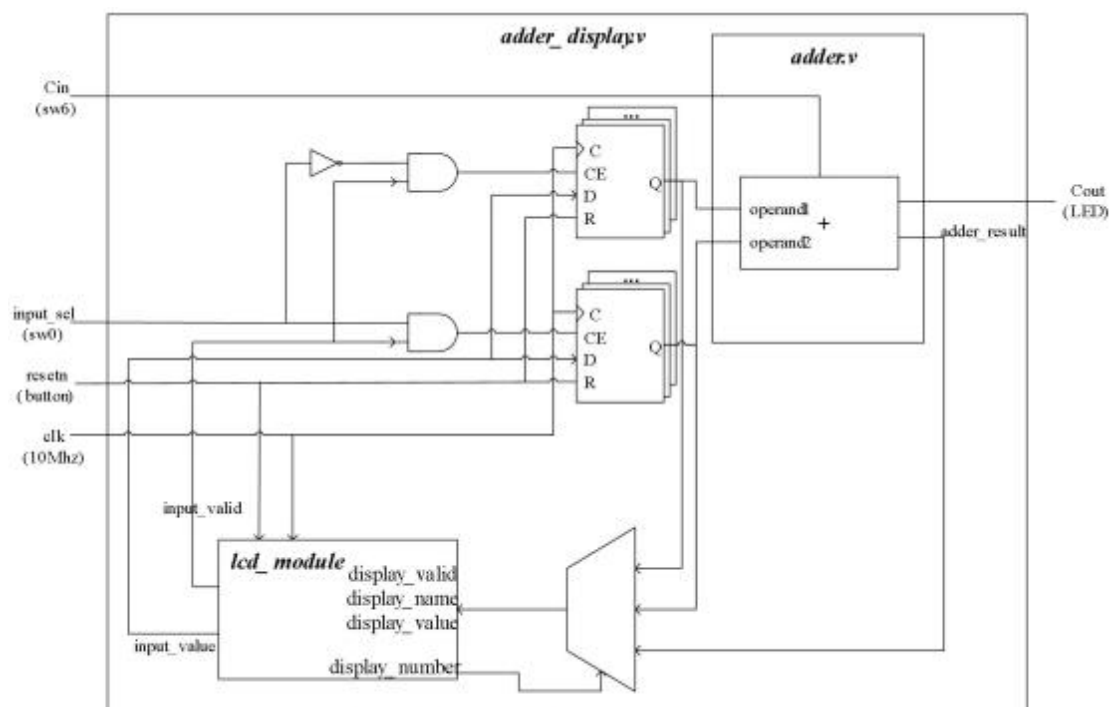


图 2.40 定点加法参考设计的顶层模块框图

$A_i$	$B_i$	$C_i$	$S_i$	$C_{i+1}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = (A_i \oplus B_i) C_i + A_i B_i$$

以上给出了有低位进位的一位加法器的真值表和函数表达式，这是加法器的基本原理。

$A_i$	$B_i$	$C_i$	$S_i$	$C_{i+1}$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = (\sim A_i) (B_i + C_i) + B_i C_i$$

以上给出了有低位借位的一位减法器的真值表和函数表达式，这是减法器的基本原理。

#### 四、实验步骤

以下给出了加法器的代码，其直接使用“+”，自动调用库里的加法器。

```

module adder(
    input [31:0] operand1,
    input [31:0] operand2,
    Input  cin,
    output [31:0] result,
    Output cout,
);
assign {cout,result} = operand1 + operand2 + cin;

Endmodule

```

代码先声明了两个加数 operand1、operand2，低位进位 cin，该位输出 result 和高位进位 cout，分别对应上述真值表中的  $A_i$ 、 $B_i$ 、 $C_i$ 、 $S_i$ 、 $C_{i+1}$ 。其后，调用加法器，输入量为前三者，输出量为后两者。

在上述代码的基础上，经过简单修改，可实现减法器。

```

module subtractor(
    input [31:0] minuend,
    input [31:0] subtrahend,
    Input  cin,
    output [31:0] result,
    Output cout,
);
assign {cout,result} = minuend - subtrahend - cin;

Endmodule

```

代码先声明了被减数 minuend，减数 subtrahend，低位借位 cin，该位输出 result 和高位借位 cout。其后，调用减法器，输入量为前三者，输出量为后两者。

## 五、实验结果分析

输入一	输入二	进/借位	理论输出	实际输出	实验截图
$(55556AAA)_{16}$	0000AAAA	0/0	5554C000	5554C000	
00000000	0000AAAA	0/1	FFFF5556	FFFF5556	
72AFT7E5	BBD27277	0/1	2E826A5C	2E826A5C	

12153524	C0895E81	1/0	D29E93A6	D29E93A6	<table><tr><th>Name</th><th>Value</th></tr><tr><td>&gt; operand1[31:0]</td><td>12153524</td></tr><tr><td>&gt; operand2[31:0]</td><td>c0895e81</td></tr><tr><td>cin</td><td>1</td></tr><tr><td>&gt; result[31:0]</td><td>d29e93a6</td></tr><tr><td>cout</td><td>0</td></tr></table>	Name	Value	> operand1[31:0]	12153524	> operand2[31:0]	c0895e81	cin	1	> result[31:0]	d29e93a6	cout	0
Name	Value																
> operand1[31:0]	12153524																
> operand2[31:0]	c0895e81																
cin	1																
> result[31:0]	d29e93a6																
cout	0																
00000000	00000000	0/0	00000000	00000000	<table><tr><th>Name</th><th>Value</th></tr><tr><td>&gt; operand1[31:0]</td><td>00000000</td></tr><tr><td>&gt; operand2[31:0]</td><td>00000000</td></tr><tr><td>cin</td><td>0</td></tr><tr><td>&gt; result[31:0]</td><td>00000000</td></tr><tr><td>cout</td><td>0</td></tr></table>	Name	Value	> operand1[31:0]	00000000	> operand2[31:0]	00000000	cin	0	> result[31:0]	00000000	cout	0
Name	Value																
> operand1[31:0]	00000000																
> operand2[31:0]	00000000																
cin	0																
> result[31:0]	00000000																
cout	0																

注：加法器中若结果大于 $(2^8-1)$ 会导致上溢，此时结果相当于真实结果加上 $(2^8-1)$ ；减法器中若被减数小于减数会导致下溢，此时结果相当于真实结果减去 $2^8$ 。

### 六、总结感想

本次实验是第一次使用 verilog 语言和 LS-CPU-EXB-002 实验箱和软件平台，虽然实验内容比较简单，但仍是学习到了很多新的知识。作为计算机组成原理实验课的第一个实验，本实验使我结合了课上学习的理论内容和操作实验的实际内容，对加法器、减法器有了更深入的了解，这为后续的学习和实验打下了基础。