Robei 集成电路实战

带符号位小数的加法设计

带符号位小数的加法设计

1. 有符号数
2. 有符号数和无符号数的说明
3. 实验设计：前端
4. 实验设计：后端

带符号位小数的加法设计

有符号数

有符号数是针对二进制来讲的。用最高位作为符号位，“0”代表“+”，“1”代表“-”；其余数位用作数值位，代表数值。

有符号数的表示：计算机中的数据用二进制表示，数的符号也只能用0/1表示。一般用最高有效位（MSB）来表示数的符号，正数用0表示，负数用1表示。

有符号数的编码方式，常用的是补码，另外还有原码和反码等。用不同二进制编码方式表示有符号数时，所得到的机器数可能不一样，但是真值应该是相同的。

带符号位小数的加法设计

数字电路中的有符号数：

可以通过关键字 signed 来定义：

```
input   signed [7:0 ] a, b;
output  signed [15:0] c;
wire    signed [15:0] x;
reg     signed [15:0] y;
```

如果没有 signed 关键字，所有数据默认都被当做无符号数。

带符号位小数的加法设计

如何确定有符号数的数值：

正数：最高位为0，其数值与相应的无符号数一样。
signed 0110 = unsigned 0110 = (dec) 6

负数：最高位为1，其绝对值等于全部数位取反加1。
1011----（全部数位取反）----0100----（+1）----0101 = 5
因此1011绝对值为5，本身的值为 -5。

带符号位小数的加法设计

带符号位的数：

最高位为独立的符号位；剩余的位数则作为无符号数表示该数的值。

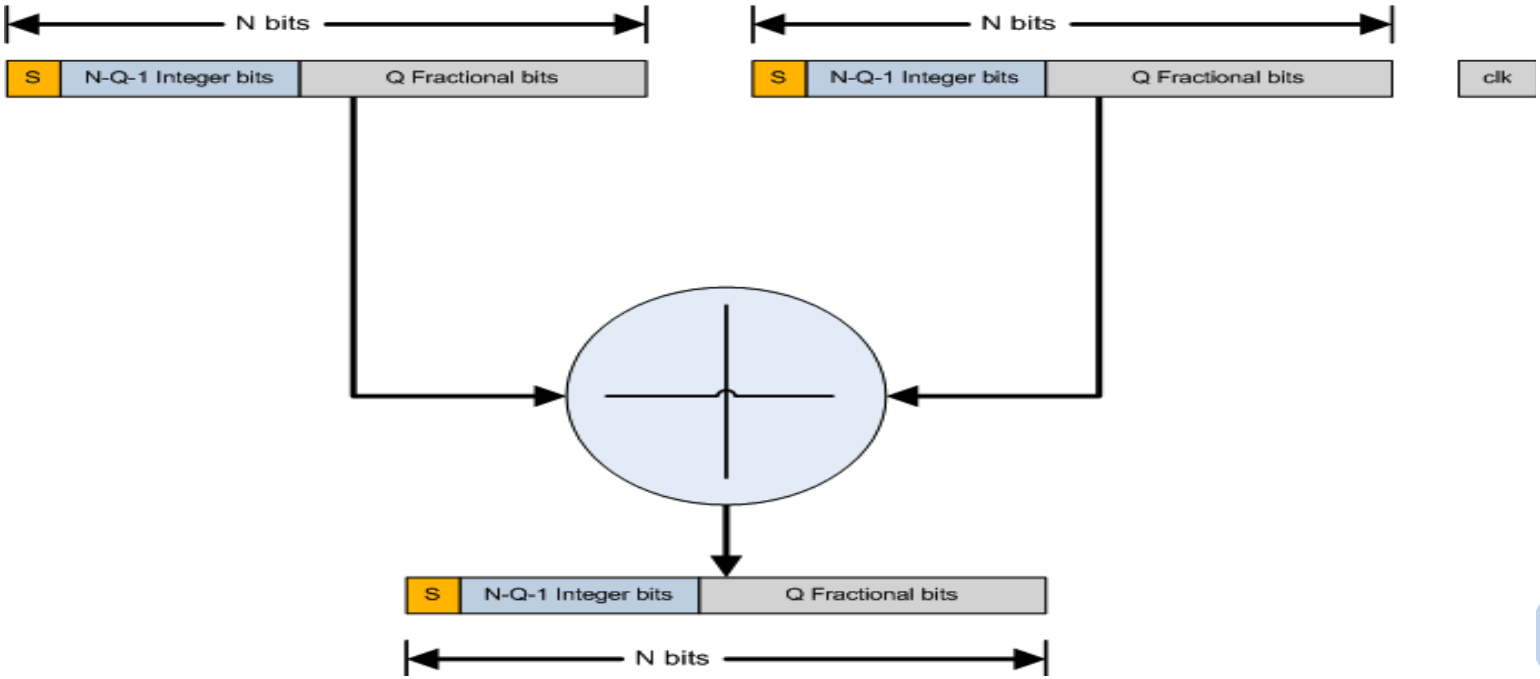
0100_1100：符号位为0，是正数；
剩余位数_100_1100作为无符号数表示76，所以该数的值为+76；

1011_0011：符号位为1，是负数；
剩余位数 _011_0011作为无符号数表示51，所以该数的值为-51。

带符号位小数的加法设计

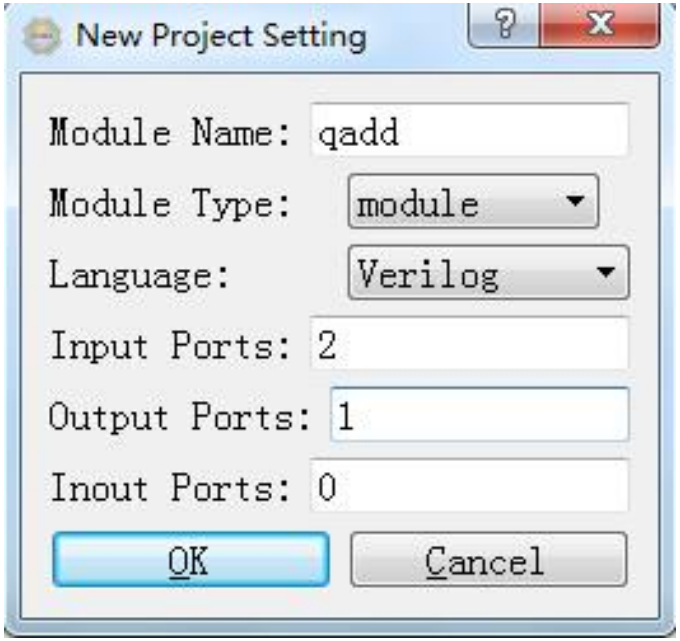
数据格式：

|1|<- N-Q-1 bits ->|<--- Q bits -->|
|S|IIIIIIIIIIIIIII|FFFFFFFFFFFFFF|



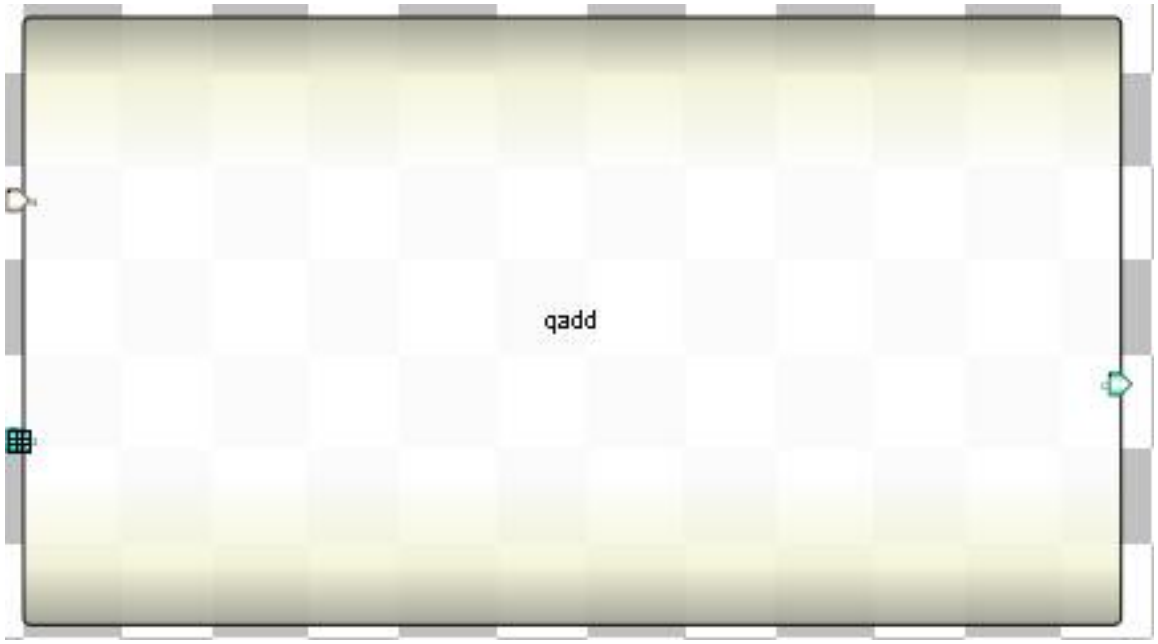
带符号位小数的加法设计

实验设计：前端设计



带符号位小数的加法设计

Name	Inout	DataType	Datasize	Function
a	input	wire	32	input a
b	input	wire	32	input b
c	output	reg	32	output c



带符号位小数的加法设计

实验代码：

```
parameter N = 4;                    //可扩展
reg [N-1:0] res;
assign c = res;

always @(a or b)
begin
    if(a[N-1] == b[N-1])            //a和b同符号
    begin
        res[N-2:0] = a[N-2:0] + b[N-2:0];
        res[N-1] = a[N-1];
    end
    else if(a[N-1] == 0 && b[N-1] == 1)
    begin                           //a为正 b为负
            if( a[N-2:0] > b[N-2:0] )
            begin                   //|a|>|b|，结果为正
          res[N-2:0] = a[N-2:0] - b[N-2:0];
              res[N-1] = 0;
            end
……
```

带符号位小数的加法设计

实验代码：

```
……
else                                    //|a|≤|b|结果为0或者负
begin
    res[N-2:0] = b[N-2:0] - a[N-2:0];
    if (res[N-2:0] == 0)        //结果为0
            res[N-1] = 0;
    else
            res[N-1] = 1;        //结果为负
end
end
else                                    //a为负 b为正
begin
    if( a[N-2:0] >= b[N-2:0] )
    begin                               //|a|≥|b|结果为0或者负
        res[N-2:0] = a[N-2:0] - b[N-2:0];
        if (res[N-2:0] == 0)
                res[N-1] = 0;  //结果为0
        else
                res[N-1] = 1;  //结果为负
    end
……
```

带符号位小数的加法设计

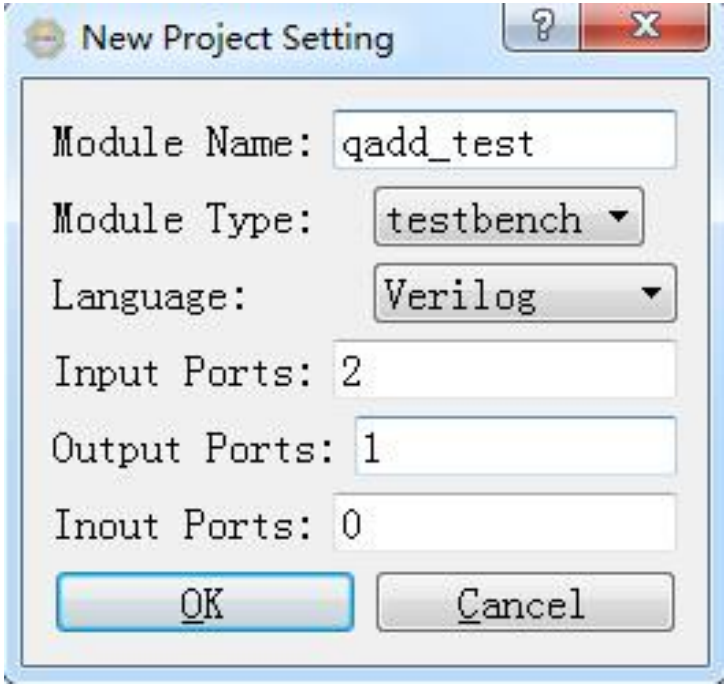
实验代码：

```
……
else                    //|a|<|b|结果为正
begin
    res[N-2:0] = b[N-2:0] - a[N-2:0];
    res[N-1] = 0;
end
end
end
```

设计完成后，保存并执行。
检查是否有错误，并生成.v文件。

带符号位小数的加法设计

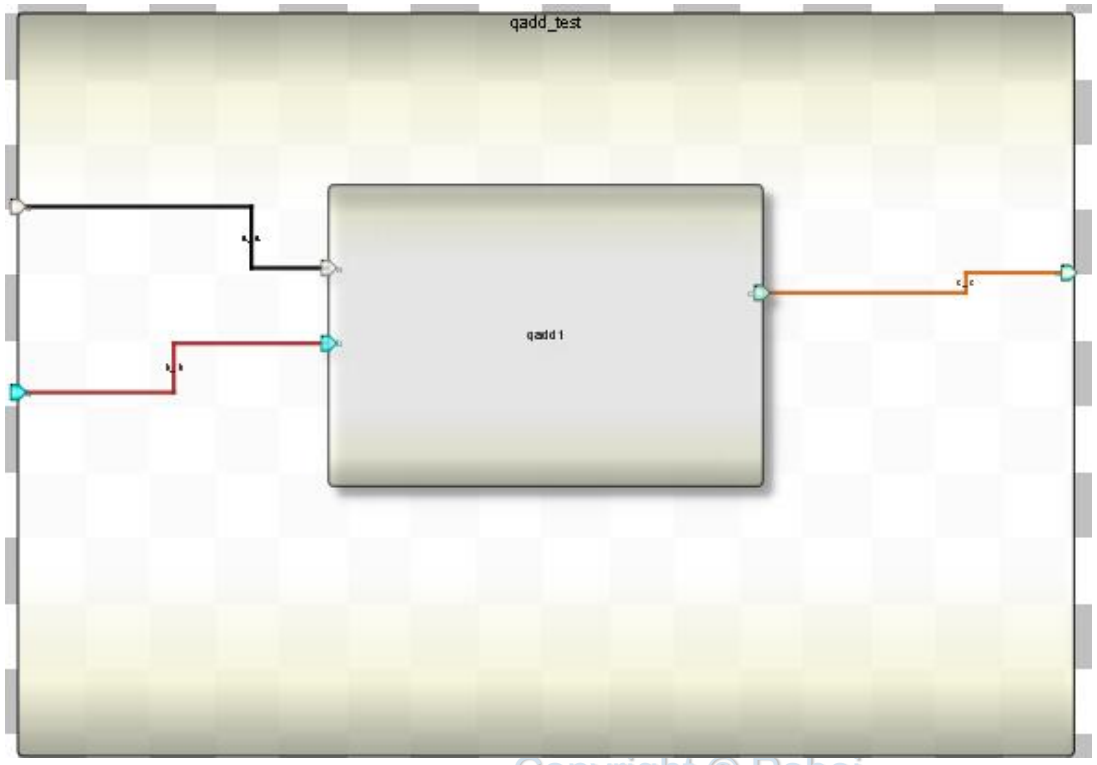
测试模块设计：



带符号位小数的加法设计

Ports:

Name	Inout	DataType	Datasize	Function
a	input	reg	32	
b	input	reg	32	
c	output	wire	32	

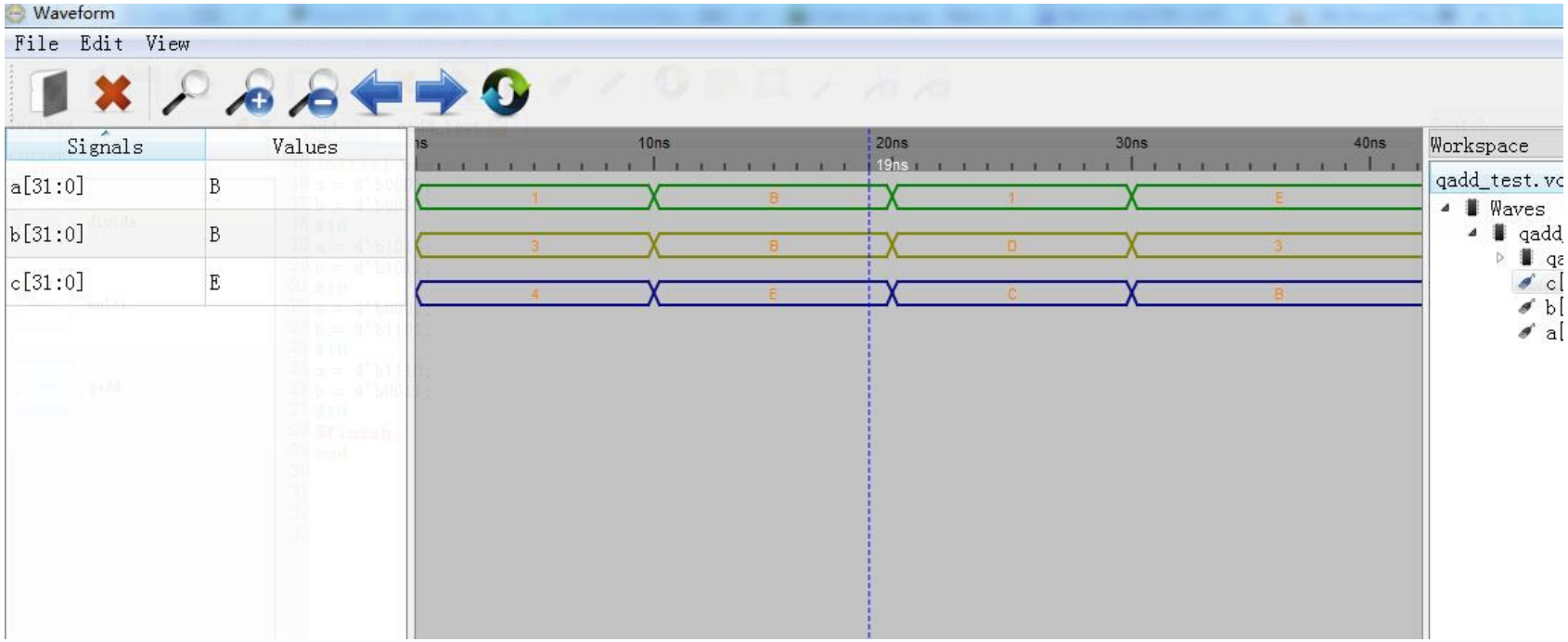


带符号位小数的加法设计

测试代码：

```
initial begin
    a = 4'b0001;
    b = 4'b0011;
#10
    a = 4'b1011;
    b = 4'b1011;
#10
     a = 4'b0001;
     b = 4'b1101;
#10
    a = 4'b1110;
     b = 4'b0011;
#10
     $finish;
end
```

带符号位小数的加法设计



① 1+3=4;

② B+B=E; 1011+1011=1110 (1)011+(1)011= -3+ -3= -6=(1)110

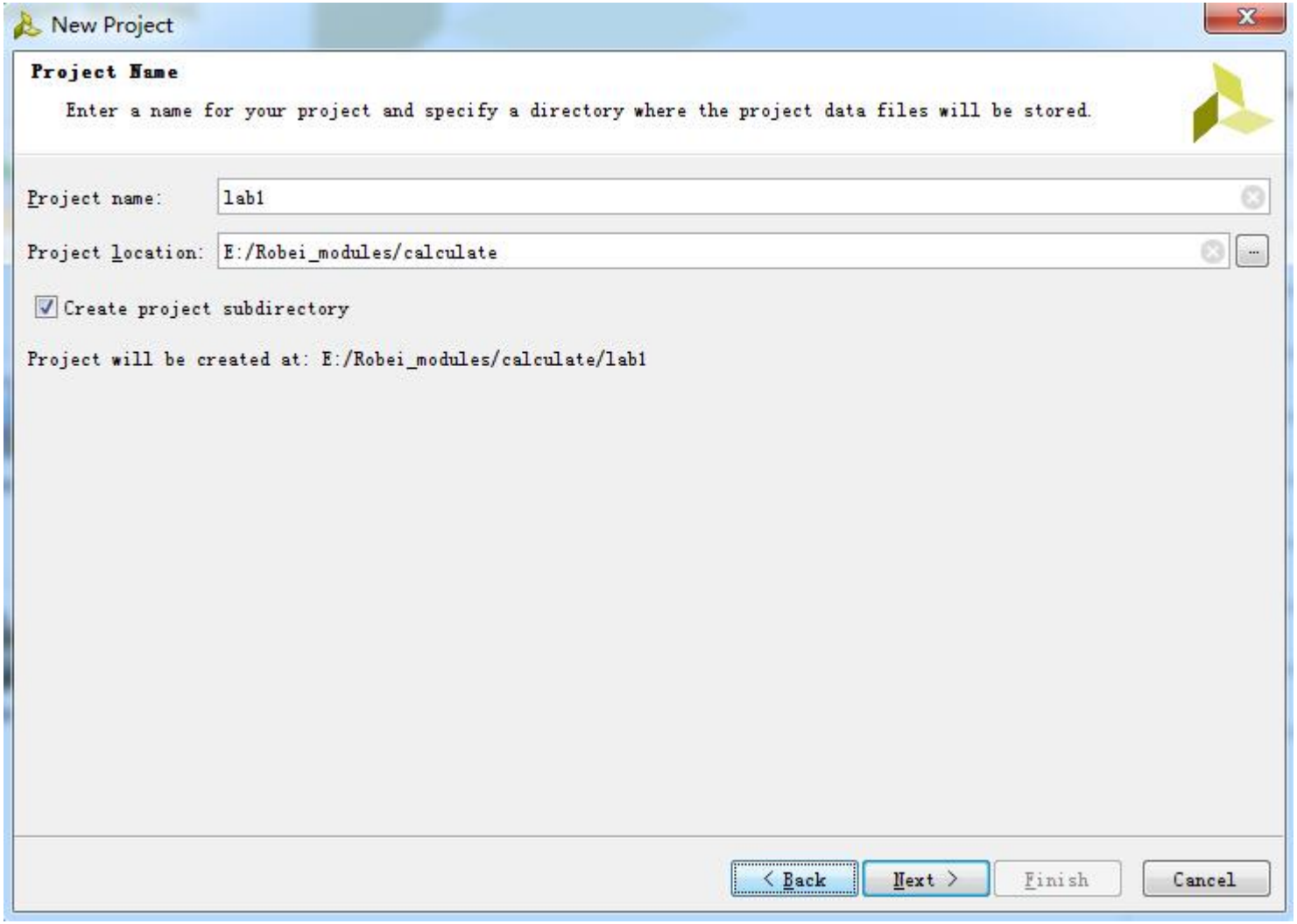
③ 1+D=C; 0001+1101=1100 (0)001+(1)101= +1+ -5= -4=(1)100

④ E+3=B; 1110+0011=1011 (1)110+(0)011= -6+ +3= -3=(1)011

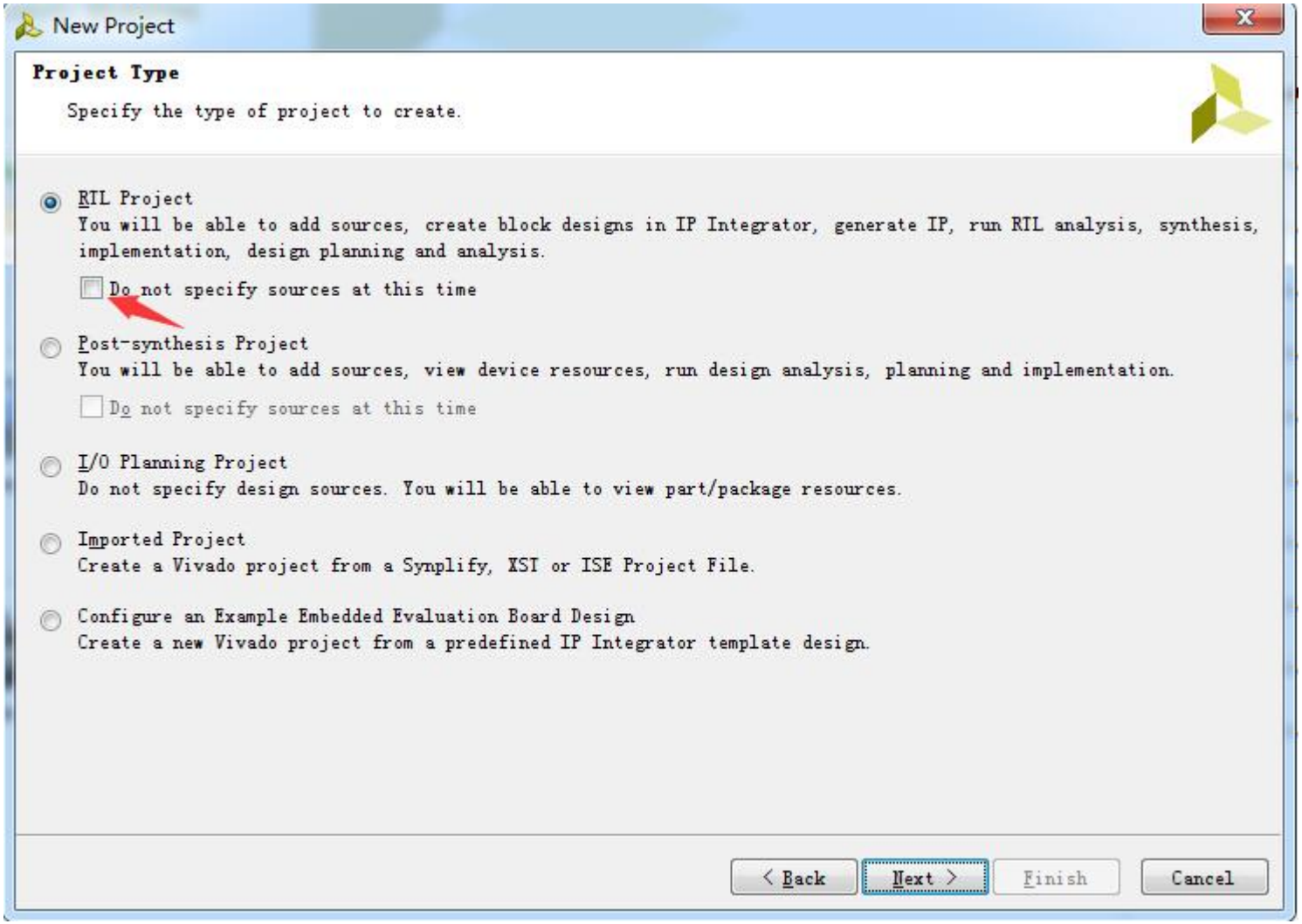
带符号位小数的加法设计

实验设计：后端

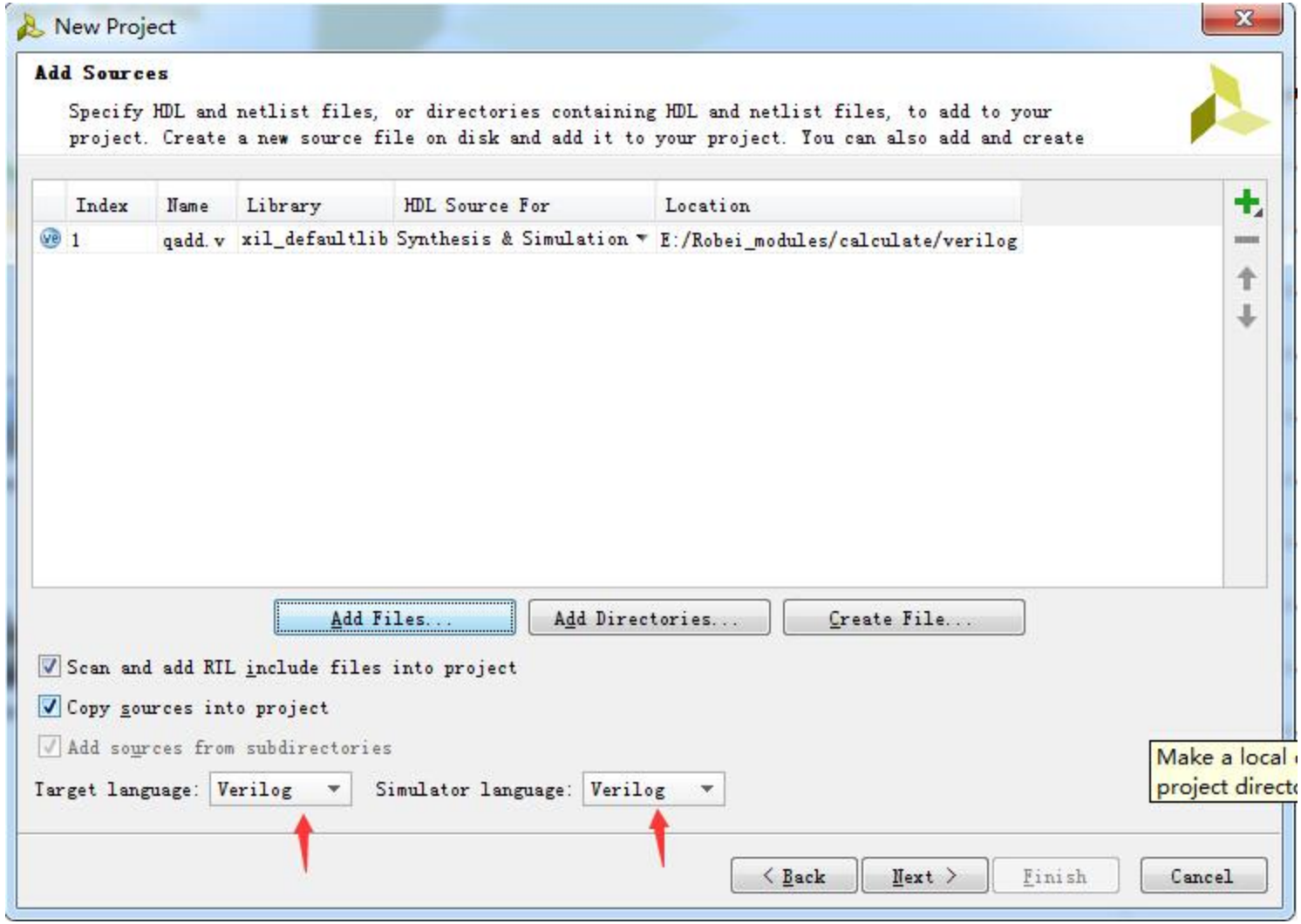
带符号位小数的加法设计



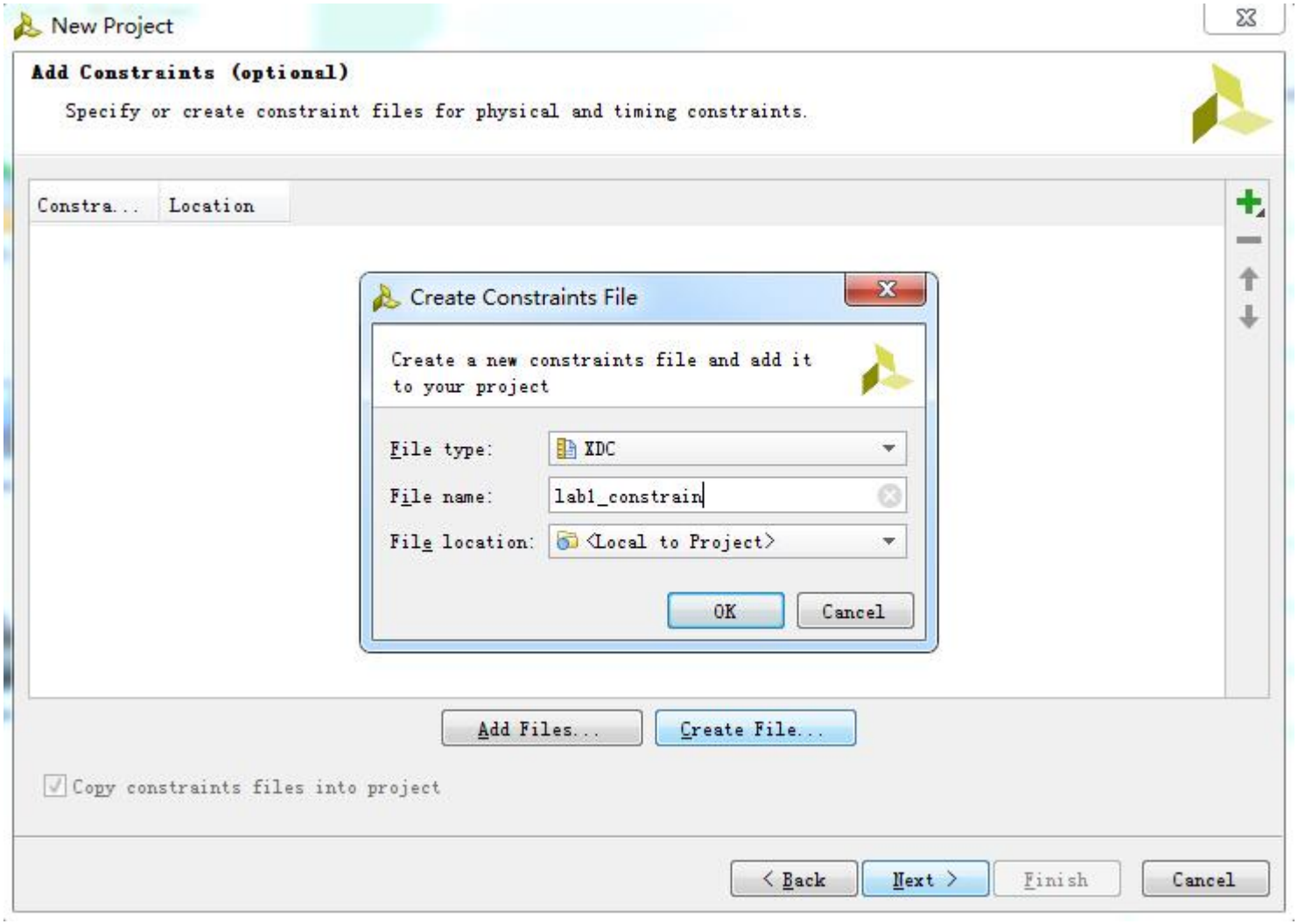
带符号位小数的加法设计



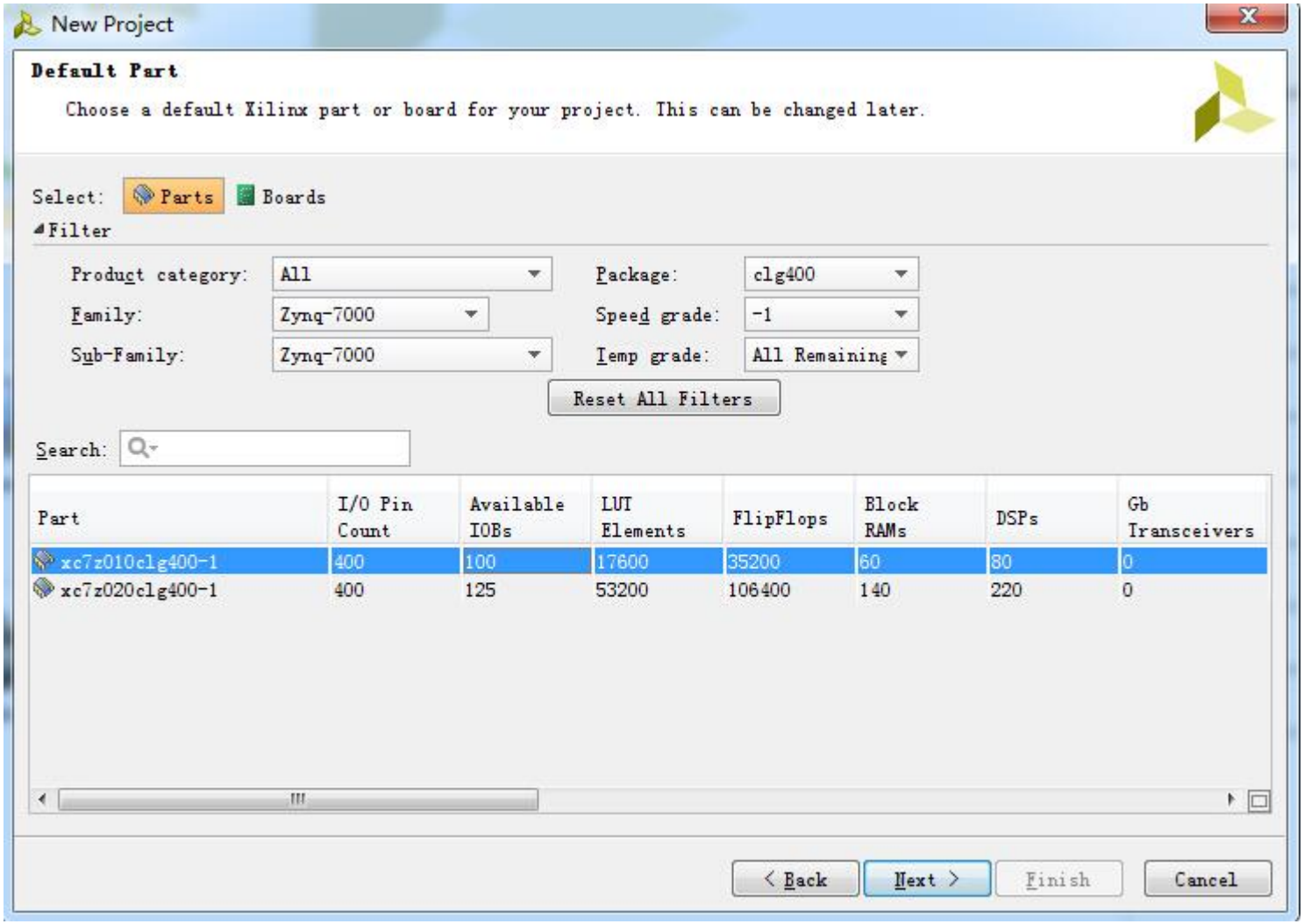
带符号位小数的加法设计



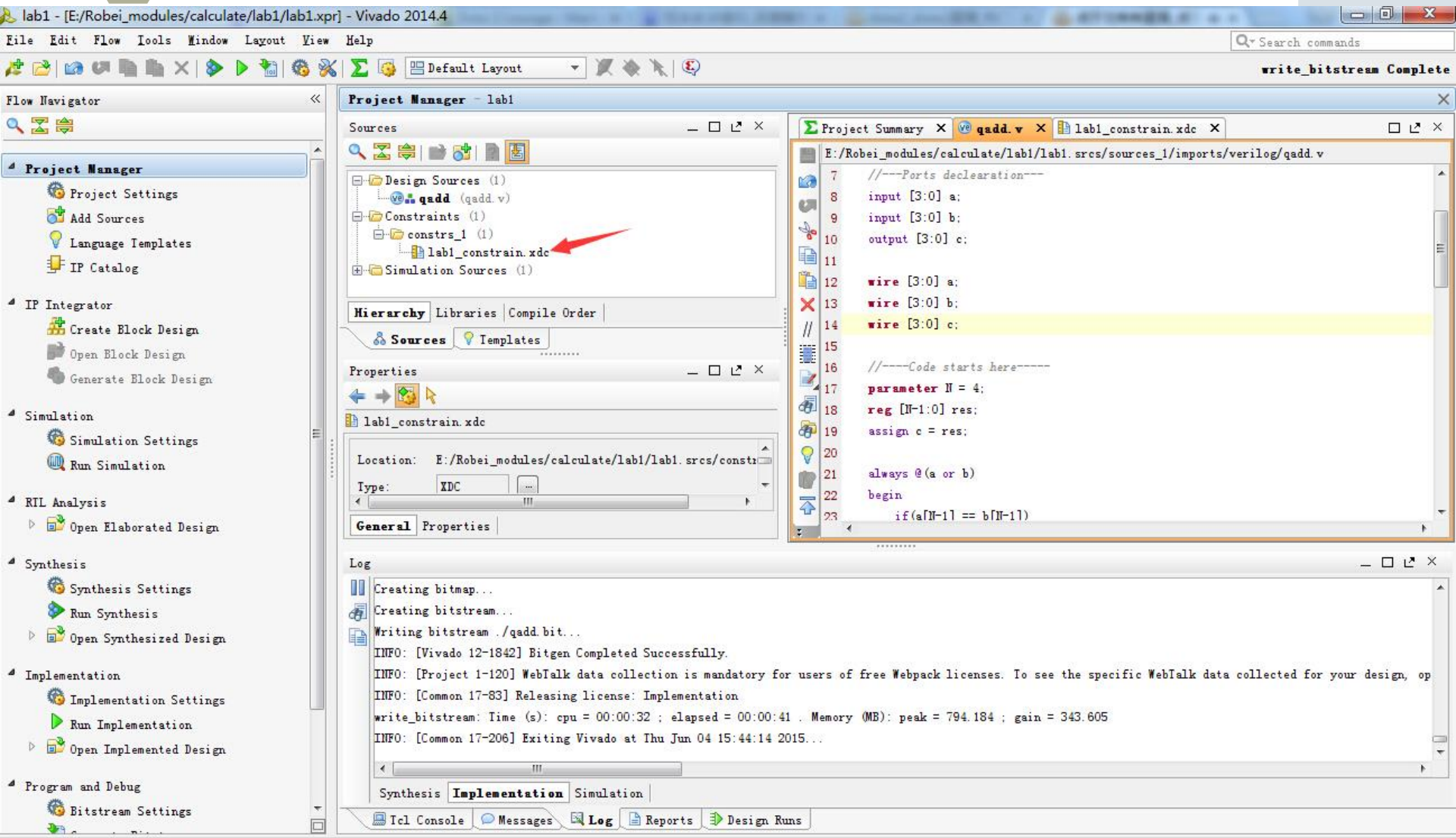
带符号位小数的加法设计



带符号位小数的加法设计



带符号位小数的加法设计

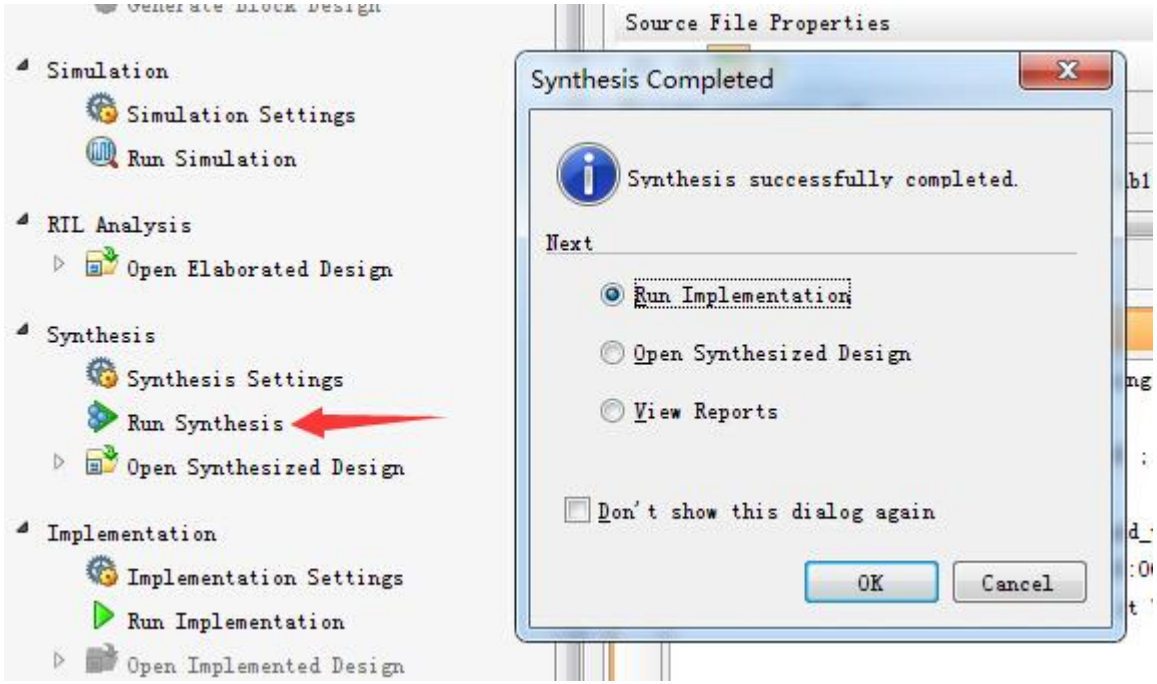


带符号位小数的加法设计

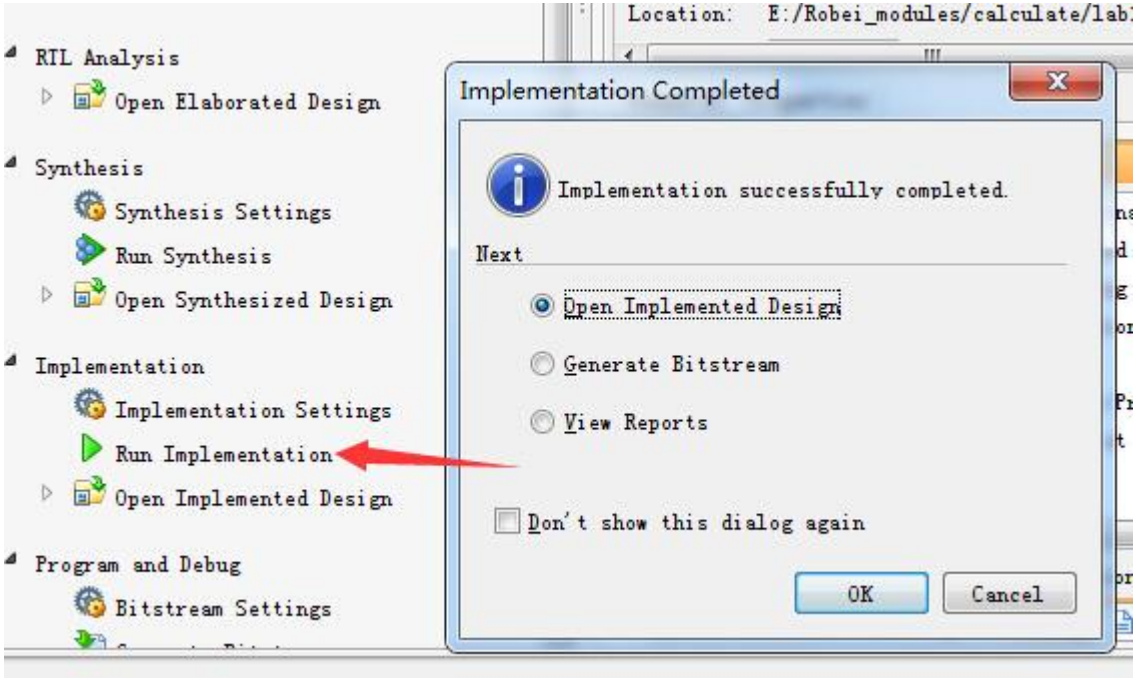
约束文件代码：

```
# ZYBO board Pins
# QADD
#PMOD JE1~4, 7~10. Switches  a[0~3], b[0~3]

set_property PACKAGE_PIN V12 [get_ports {a[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {a[0]}]

set_property PACKAGE_PIN W16 [get_ports {a[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {a[1]}]

set_property PACKAGE_PIN J15 [get_ports {a[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {a[2]}]

set_property PACKAGE_PIN H15 [get_ports {a[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {a[3]}]
……
……
```

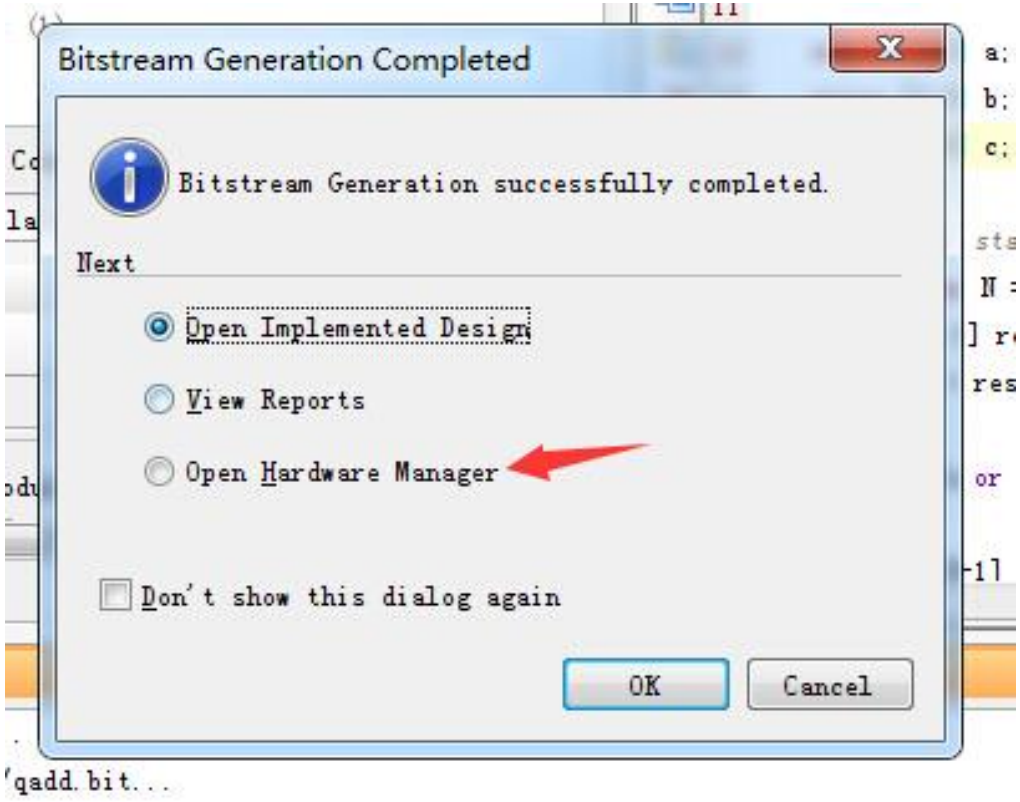
带符号位小数的加法设计



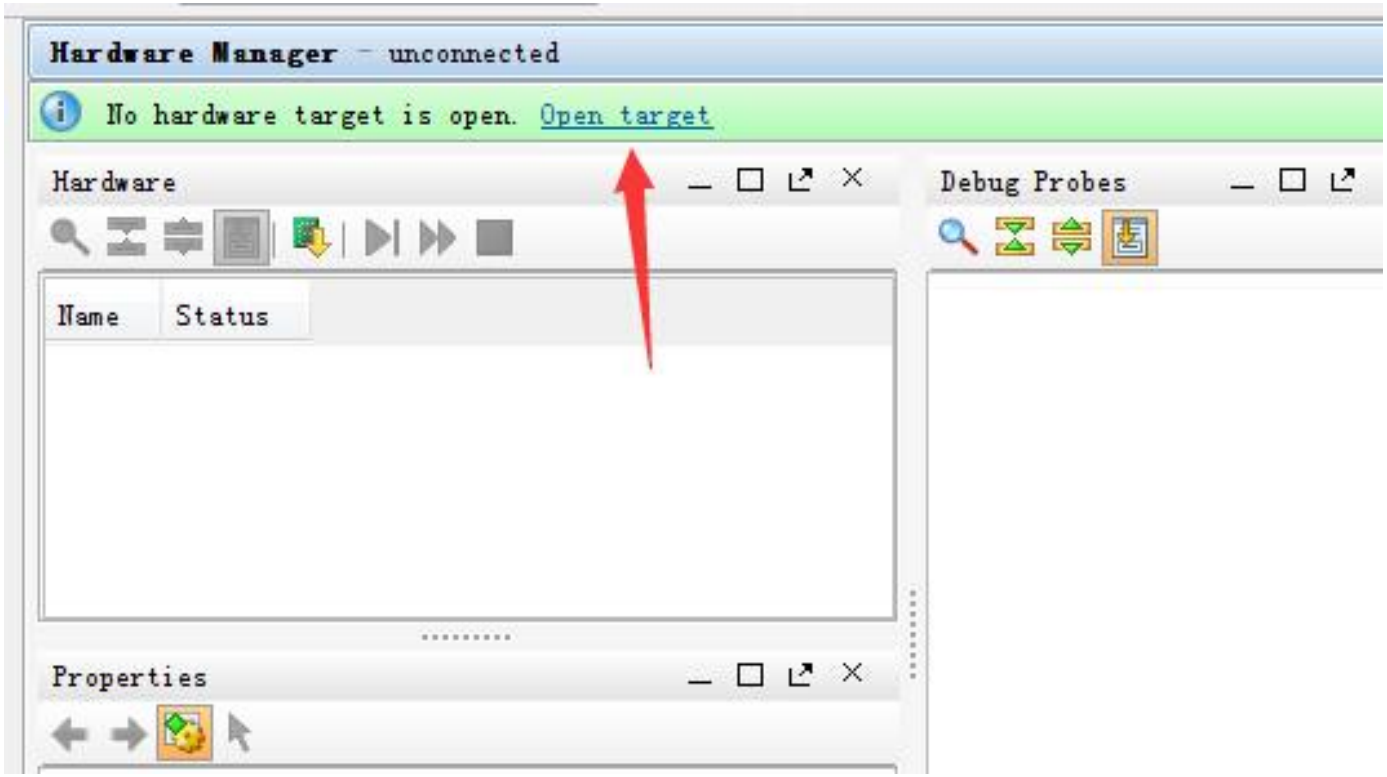
带符号位小数的加法设计



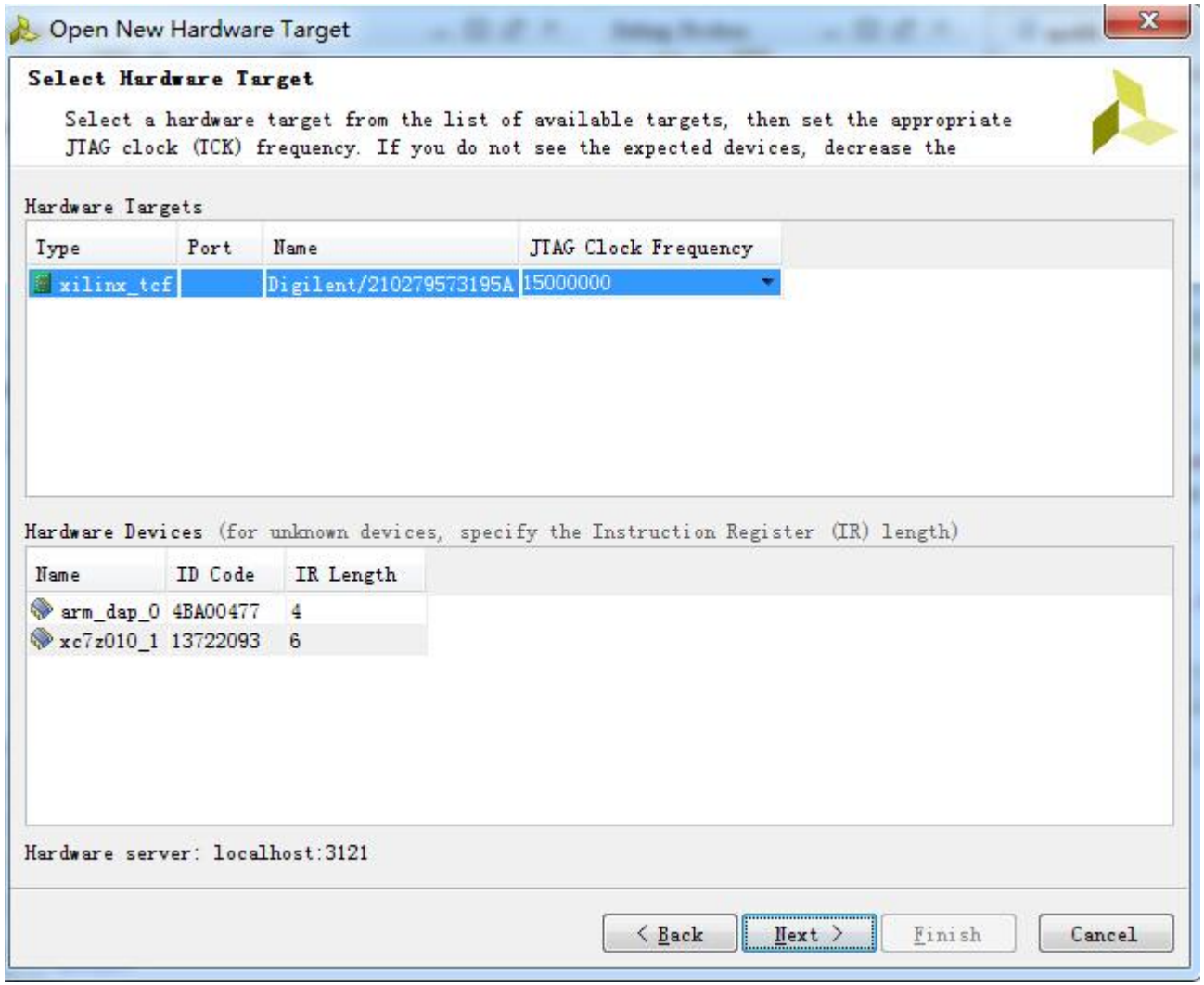
带符号位小数的加法设计



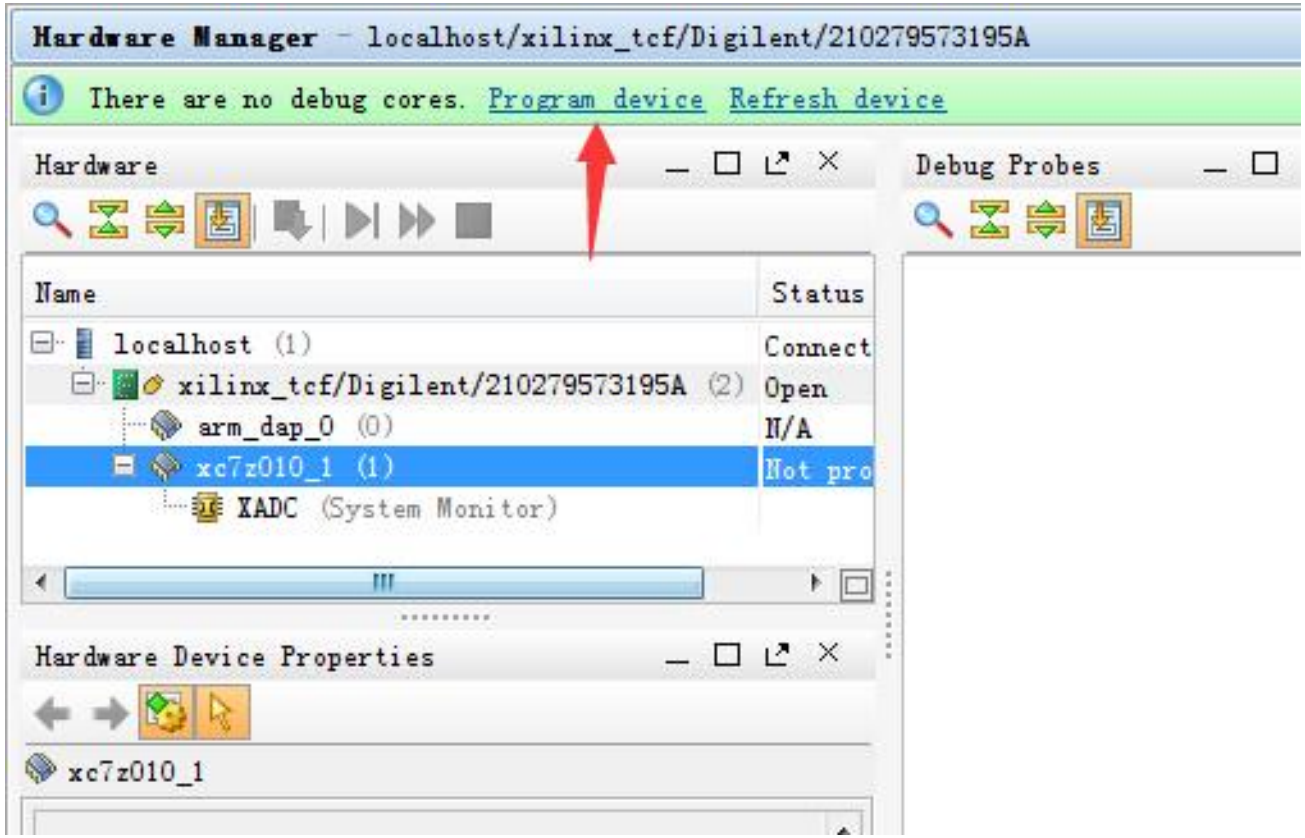
带符号位小数的加法设计



带符号位小数的加法设计



带符号位小数的加法设计



www.robei.com

Thank You!