

**计算机与信息 学院实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验课程： | 计算机组成原理实验 | | | | |
| 实验编号： | Project 6 | | | | |
| 实验名称： | 数据的机器级表示 | | | | |
| 实验人员： | 学号 | 18111303044 | | | |
| 姓名 | 邵一波 | | | |
| 班级 | 2018级计算机类一班 | | | |
| 实验日期： | 2019-10-21 | | | | |
| 实验室： | 学苑南楼2幢202室 | | | | |
|  |  | | | | |
| 实验评价： |  | | | | |
| 实验成绩： | |  | 评价日期： |  |
|  | 指导教师： | |  | | |

# 数据的机器级表示

# 一、实验目的

## 1.掌握定点数的原码、补码、反码、移码表示方法。

## 2.掌握浮点数的IEEE754表示方法。

# 二、实验工具

## 1.Xilinx Vivado 2014.2软件。

## 2.Visual Studio 2010软件。

## 3.Windows系统PC机。

# 三、实验要求

## 1.利用Verilog HDL的完成“定点整数编码转换逻辑”建模和验证。

## 2.利用C语言完成平台数据类型的验证。

# 四、实验内容

## 1. “定点整数编码转换逻辑”建模与验证

表1 定点整数的编码方法

|  |  |  |
| --- | --- | --- |
| 序号 | 编码 | 英文 |
| 1 | 原码 | Code |
| 2 | 反码 | One's complement |
| 3 | 补码 | Two's complement |
| 4 | 移码 | Offset binary |

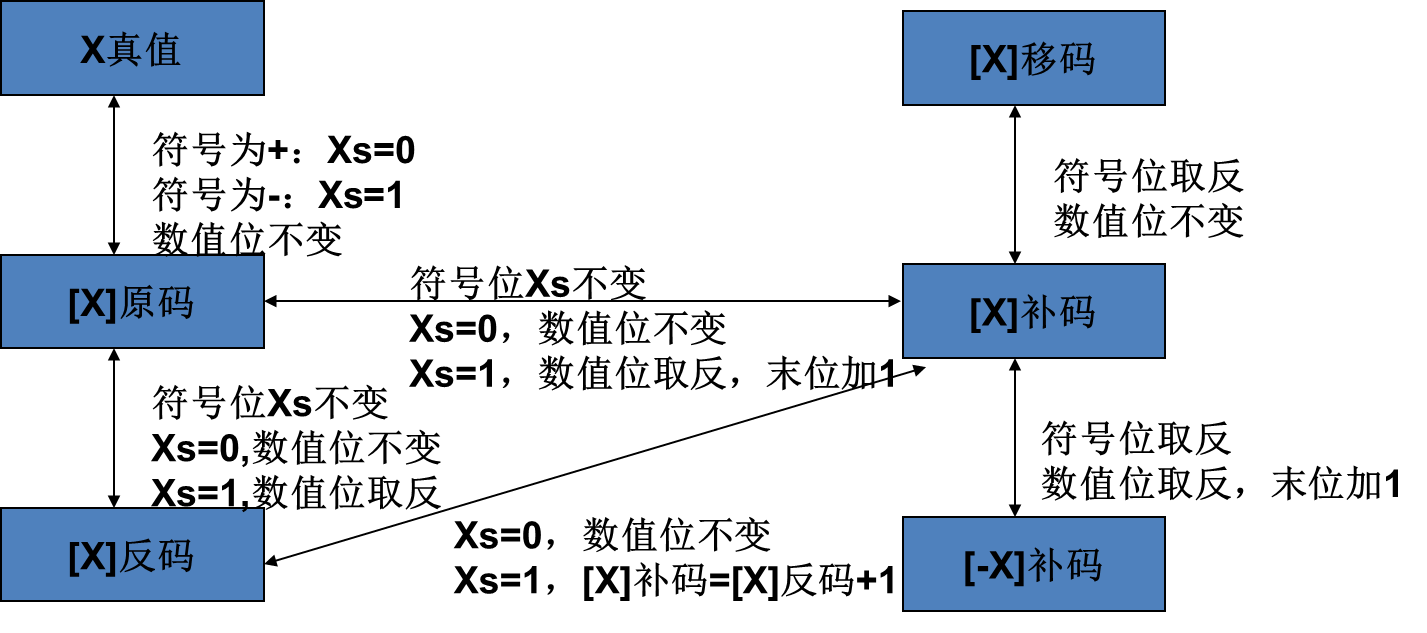


图1 各种码制之间的关系及转换方法

给定一个32位的原码表示的定点整数，设计逻辑电路完成原码→反码，反码→补码，补码→移码，并进行仿真、逻辑分析、电路综合。

例如，原码→反码转换逻辑如下。

module Code2OnesComplement (N,M);//原码->反码

parameter WIDTH=32;

input wire [WIDTH-1:0] N;

output wire [WIDTH-1:0] M;

assign M=N[WIDTH-1]?{N[WIDTH-1],~N[WIDTH-2:0]}:N;

//assign M=N[WIDTH-1]?{N+1}:N;

endmodule

仿真模块如下。

module test();//simulation

parameter WIDTH=32;

reg [WIDTH-1:0] N;

wire [WIDTH-1:0] M;

Code2OnesComplement u0(N,M);

initial

begin

#10;

N=32'h00\_00\_00\_01;

#10;

N=32'h00\_00\_00\_02;

#10;

N=32'h80\_00\_00\_01;

#10;

N=32'h80\_00\_00\_02;

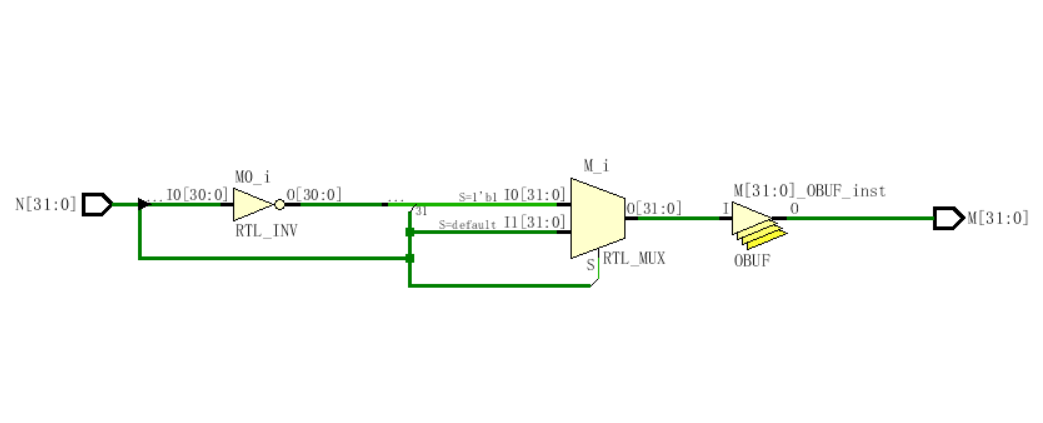
#10;

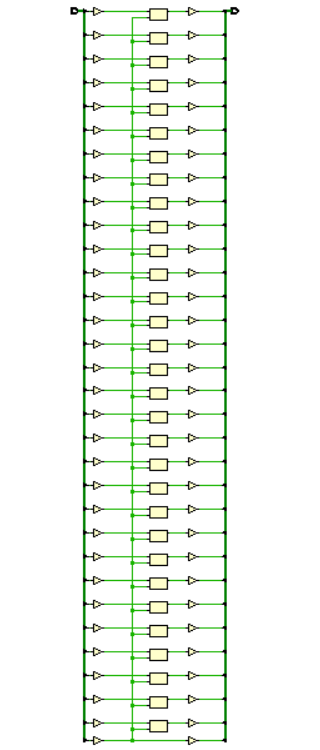
$finish;

end

endmodule







反码→补码转换逻辑如下。

module OnesComplement2TwosComplement (N,M);

parameter WIDTH=32;

input wire [WIDTH-1:0] N;

output wire [WIDTH-1:0] M;

assign M=N[WIDTH-1]?(N+1):N;

endmodule

module OnesComplement2TwosComplement (N,M);

parameter WIDTH=32;

input wire [WIDTH-1:0] N;

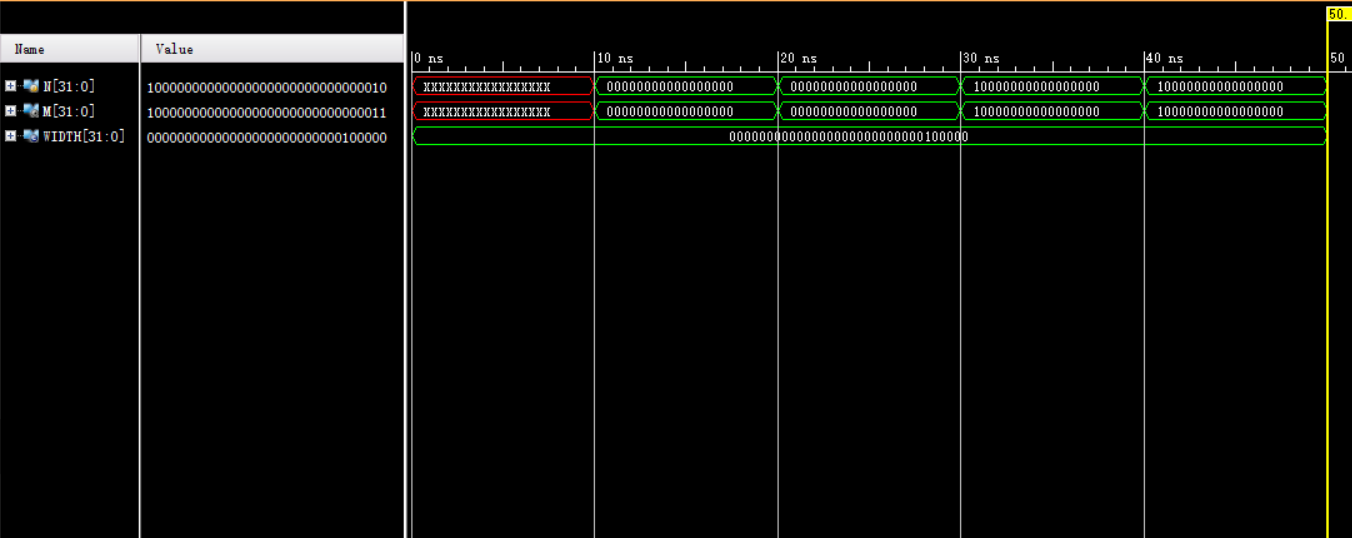
output wire [WIDTH-1:0] M;

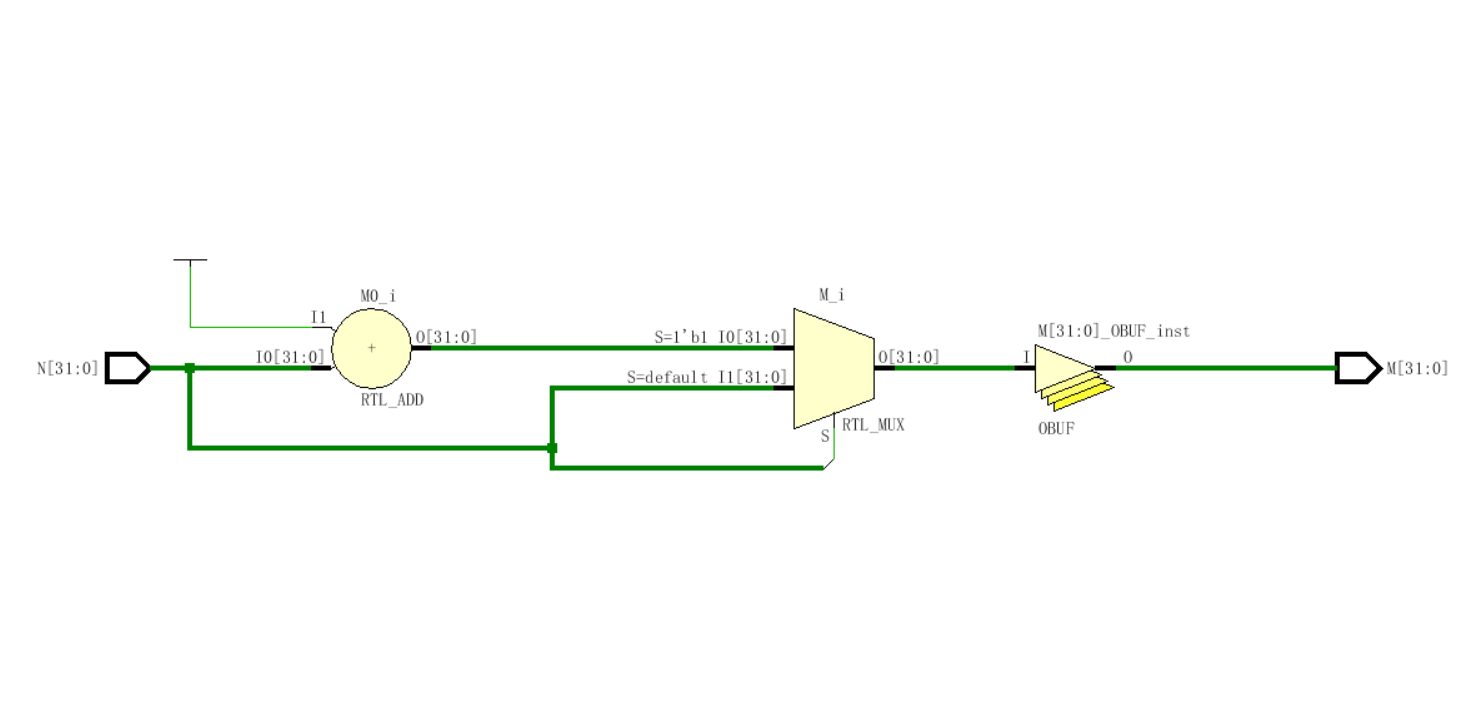
wire cout;

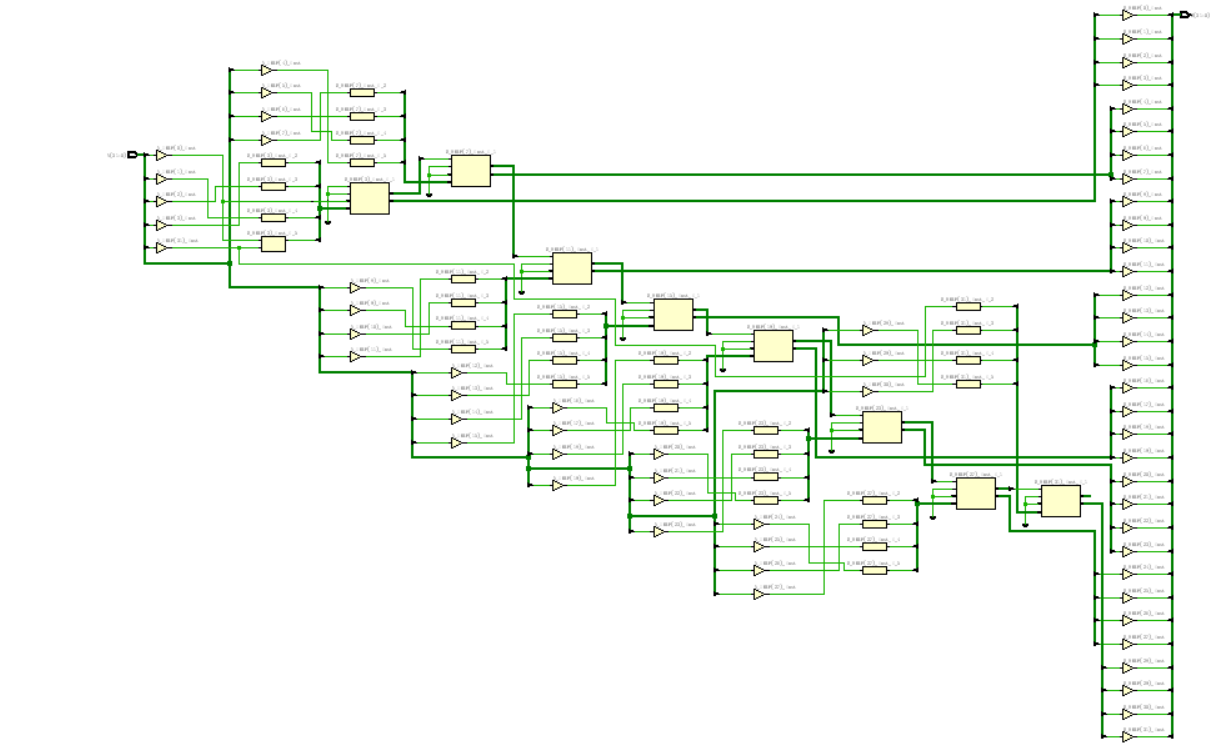
assign {cout,X} = N+32’d1;

assign M=N[WIDTH-1]?X:N;

endmodule







补码→移码转换逻辑如下。

module TwosComplement2Offset (N,M);

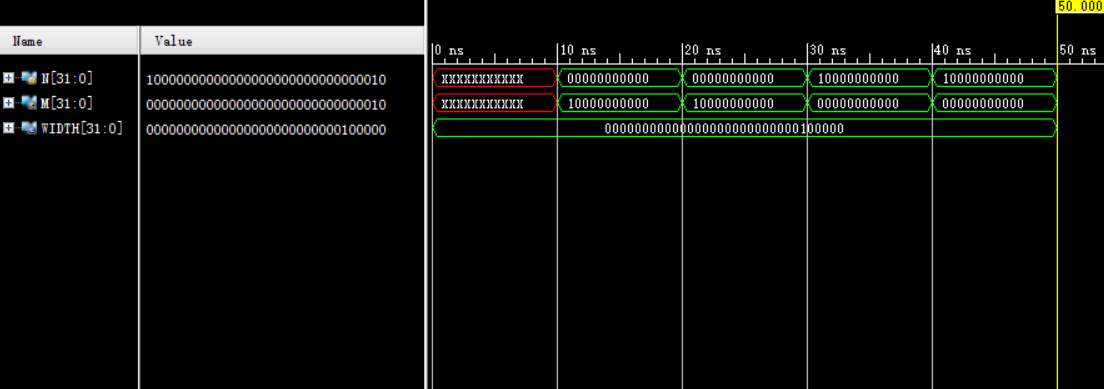
parameter WIDTH=32;

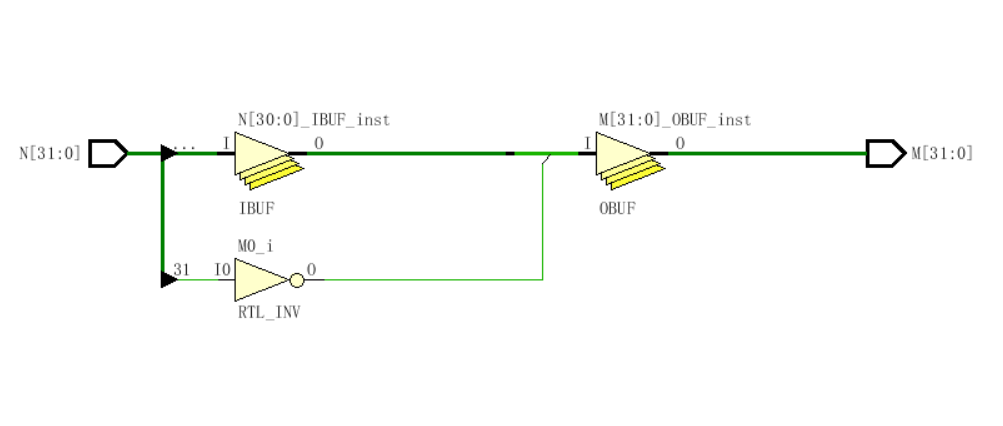
input wire [WIDTH-1:0] N;

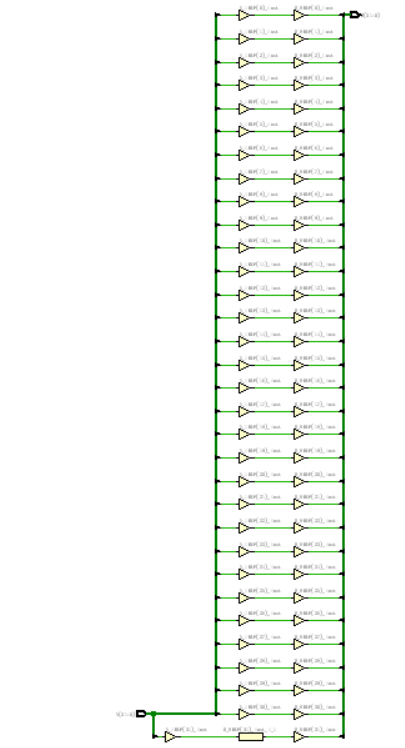
output wire [WIDTH-1:0] M;

assign M={~N[WIDTH-1],N[WIDTH-2:0]};

endmodule







## 2. C语言中的整数数据类型

调试分析如下程序。

void decimal2binary(unsigned int x, char \*s)

{

unsigned int d;

short int i=31;

while (i>=0)

{

d=x%2;

if(d)

{

s[i]='1';

//printf("%c",s[i]);

}

else

{

s[i]='0';

//printf("%c",s[i]);

}

x=x/2;

i--;

}

}

int main()

{

char s[33];

long int x=-1;

unsigned int u=2147483648;

s[32]='\0';

printf("x=%u=%d=0x%x\n",x,x,x);

printf("converting decimal to binary...\n");

decimal2binary(x,s);

printf("%s\n",s);

printf("converting end\n");

printf("\n");

printf("u=%u=%d=0x%x\n",u,u,u);

printf("converting decimal to binary...\n");

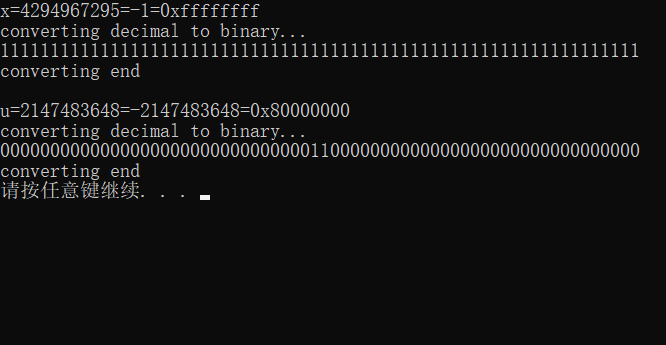
decimal2binary(u,s);

printf("%s\n",s);

printf("converting end\n");

return 0;

}



## 3. C语言中的浮点数据类型

调试分析如下程序。

union var{

long int l;

float f;

};

void show\_float(float x)

{

var v;

v.f=x;

printf("Float number %f=0x%x\n",v.f,v.l);

}

void show\_floatpointer(float \* x)

{

printf("Float point %f, address=0x%x\n",x,&x);

}

int main()

{

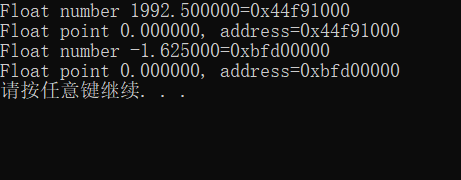
float x=1992.5,y=-1.625;

show\_float(x); show\_floatpointer(&x);

show\_float(y); show\_floatpointer(&y);

return 0;

}



# 五、实验思考

## 1. 已知某数的补码为，如何设计设计逻辑转换电路获得的补码？

**【答】**

**各位取反，末尾加一。**

**module trans(N,M);**

**parameter WIDTH=32;**

**input wire [WIDTH-1:0] N;**

**output wire [WIDTH-1:0] M;**

**assign M={~N[WIDTH-1],N[0]+1};**

**endmodule**

## 2. 以下是一个C语言程序，用来计算一个数组a中每个元素的和。当参数len为0时，返回值应该是多少？请问这是什么原因造成的，并说明程序应该如何修改。

float sum\_elements(float a[], unsigned len)

{

int i;

float result = 0;

if(len!=0&&len<5)

for (i = 0; i <= len–1; i++)

result += a[i];

return result;

}

**【答】返回值应该是0，因为unsigned是无符号类型，而int是有符号类型。应该将unsigned改为int。**

## 3. 下面这段代码可以用来测试一下你的编译器是大端模式还是小端模式。

void main()

{

short int x;

char x0,x1;

x = 0x1122;

x0 = ((char \*)&x)[0];//低地址单元

x1 = ((char \*)&x)[1];//高地址单元

printf("[%x] = %x\n ",&x,x);

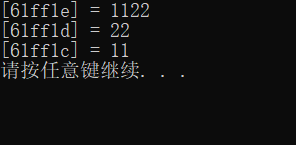
printf("[%x] = %x",&x0,x0);

printf("[%x] = %x",&x1,x1);

return;

}

**【答】**



**故为小端模式**