实验报告

智能 1602 班 201608010623 李路

实验名称:

RISC-V 基本指令集汇编器设计与实现

实验目标:

设计一个 RISC-V 基本整数指令汇编器，能够实现将汇编指令转化为二进制

实验要求 :

采用 C/C++编写程序 汇编器的输入是进行模拟的汇编指令，汇编器的输出是经过汇编后的二进制指令文件

实验内容 :

1. 汇编器简介

汇编器是将汇编语言翻译为机器语言的程序。一般而言，汇编生成的是目标代码，需要经链接器生成可执行代码才可以执行

汇编语言是一种以处理器指令系统为基础的低级语言，采用助记符表达指令操作码，采用标识符表示指令操作数。作为一门语言，对应于高级语言的编译器，需要一个汇编器将汇编语言原文件汇编成机器可执行的代码。

2. RISC-V 指令集

RV32I 指令集包含了六种基本指令格式，分别是:

R 类型指令：用于寄存器到寄存器操作

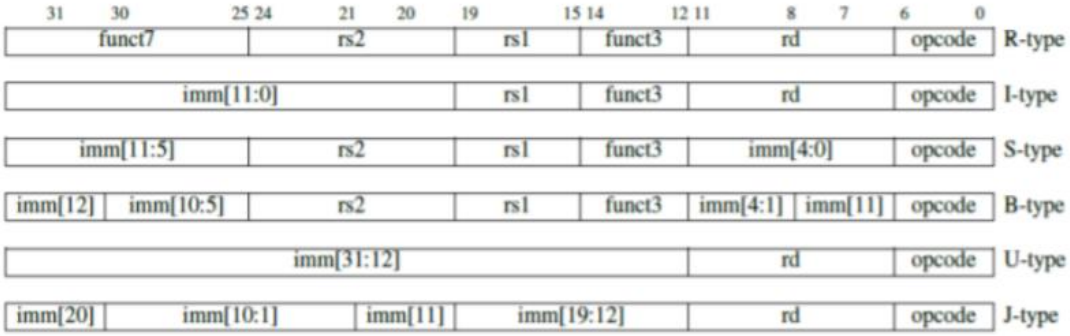
I 类型指令：用于短立即数和访存 load 操作

S 类型指令：用于访存 store 操作

B 类型指令：用于条件跳转操作

U 类型指令：用于长立即数

J 类型指令：用于无条件跳转



3. RISC-V 指令集

Instruction	Constraints	Code Points	Purpose
LUI	*rd*=x0	2^{20}	*Reserved for future standard use*
AUIPC	*rd*=x0	2^{20}	
ADDI	*rd*=x0, and either *rs1*≠x0 or *imm*≠0	$2^{17}-1$	
ANDI	*rd*=x0	2^{17}	
ORI	*rd*=x0	2^{17}	
XORI	*rd*=x0	2^{17}	
ADDIW	*rd*=x0	2^{17}	
ADD	*rd*=x0	2^{10}	
SUB	*rd*=x0	2^{10}	
AND	*rd*=x0	2^{10}	
OR	*rd*=x0	2^{10}	
XOR	*rd*=x0	2^{10}	
SLL	*rd*=x0	2^{10}	
SRL	*rd*=x0	2^{10}	
SRA	*rd*=x0	2^{10}	
ADDW	*rd*=x0	2^{10}	
SUBW	*rd*=x0	2^{10}	
SLLW	*rd*=x0	2^{10}	
SRLW	*rd*=x0	2^{10}	
SRAW	*rd*=x0	2^{10}	
FENCE	*pred*=0 or *succ*=0	$2^{5}-1$	
SLTI	*rd*=x0	2^{17}	*Reserved for custom use*
SLTIU	*rd*=x0	2^{17}	
SLLI	*rd*=x0	2^{11}	
SRLI	*rd*=x0	2^{11}	
SRAI	*rd*=x0	2^{11}	
SLLIW	*rd*=x0	2^{10}	
SRLIW	*rd*=x0	2^{10}	
SRAIW	*rd*=x0	2^{10}	
SLT	*rd*=x0	2^{10}	
SLTU	*rd*=x0	2^{10}	

汇编器程序框架：

```
void getnum(string s){
    memset(d,0,sizeof(d));
    int j=0;
    for(int i=0;i<s.length();i++){
        if(s[i]>='0'&&s[i]<='9'){
            d[j] = d[j]*10+int(s[i]-'0');
            if(i+1<s.length()&&(s[i+1]<'0' || s[i+1]>'9')) j++;
        }
    }
```

```
}
int main(){
	//读入输入文件
	freopen("input.txt","r",stdin);

	presolve();
	string op,s;
	//将结果输入输出文件
	//freopen("output.txt","w",stdout);
	ofstream outFile("out.dat",ios::out | ios::binary);
	while(cin>>op>>s){
		//进行译码过程
		if(){
				getnum(s);
				//输出译码结果
				//solveR(opcode[op],d[0],d[1],d[2]);
				//写入 txt 中
		}
		else if(){
				getnum(s);
				//输出译码结果
				//solveR(opcode[op],d[0],d[1],d[2]);
				//写入 txt 中
		}

	fclose(stdin);
	outFile.close();
	//fclose(stdout);

	return 0;
}
```

测试:

部件	配置
CPU	core i7-6300U
内存	8GB
操作系统	windows 10

测试记录：

用于输入的文件如下：

```
ADD r3,r1,r2
SUB r3,r1,r2
XOR r3,r1,r2
OR r3,r1,r2
AND r3,r1,r2
SLL r3,r1,r2
SRL r3,r1,r2
SRA r3,r1,r2
SLT r3,r1,r2
SLTU r3,r1,r2

LB r2,r1,10
LH r2,r1,10
LW r2,r1,10
LBU r2,r1,10
LHU r2,r1,10
ADDI r2,r1,10
SLTI r2,r1,10
SLTIU r2,r1,10
XORI r2,r1,10
ORI r2,r1,10
ANDI r2,r1,10
SLLI r2,r1,10
SRLI r2,r1,10
SRAI r2,r1,10

SB r1,r2,36
SH r1,r2,36
SW r1,r2,36

LUI r1,200
AUIPC r1,200

BEQ r1,r2,200
BNE r1,r2,200
BLT r1,r2,200
BGE r1,r2,200
BLTU r1,r2,200
BGEU r1,r2,200
```

JAL r1,100
JALR r2,r1,100
输出： 输出的结果存储在 output.txt 里
00000000001000001000000110110011
01000000001000001000000110110011
00000000001000001100000110110011
00000000001000001110000110110011
00000000001000001000000110110011
00000000001000001001000110110011
00000000001000001101000110110011
01000000001000001101000110110011
00000000001000001010000110110011
00000000001000001011000110110011
00000000101000001000000100000011
00000000101000001001000100000011
00000000101000001010000100000011
00000000101000001100000100000011
00000000101000001101000100000011
00000000101000001000000100010011
00000000101000001010000100010011
00000000101000001011000100010011
00000000101000001100000100010011
00000000101000001110000100010011
00000000101000001000000100010011
00000000101000001001000100010011
00000000101000001101000100010011
01000000101000001101000100010011
00000010001000001000001000100011
00000010001000001001001000100011
00000010001000001010001000100011
00000000000011001000000010110111
00000000000011001000000010010111
00011000001000001000100001100011
00011000001000001001100001100011
00011000001000001100100001100011
00011000001000001101100001100011
00011000001000001110100001100011
00011000001000001111100001100011
00001100100000000000000011101111
00000110010000001000000101100111

分析和结论：

从测试结果可以看出编写的汇编器代码确实能够将汇编代码编译成二进制结果，可以说明编

写的汇编器代码正确，达到了实验的目的。