实验报告

计科 1605 班 201608010622 符希健

实验目标:

RISC-V 基本指令集汇编器设计与实现

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. RISC-V 指令集编码格式

RV32I Base Instruction Set

imm[31:12]				rd	0110111	LUI
imm[31:12]				rd	0010111	AUIPC
imm[20|10:1|11|19:12]				rd	1101111	JAL
imm[11:0]		rs1	000	rd	1100111	JALR
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ
imm[12|10:5]	rs2	rs1	001	imm[4:1|11]	1100011	BNE
imm[12|10:5]	rs2	rs1	100	imm[4:1|11]	1100011	BLT
imm[12|10:5]	rs2	rs1	101	imm[4:1|11]	1100011	BGE
imm[12|10:5]	rs2	rs1	110	imm[4:1|11]	1100011	BLTU
imm[12|10:5]	rs2	rs1	111	imm[4:1|11]	1100011	BGEU
imm[11:0]		rs1	000	rd	0000011	LB
imm[11:0]		rs1	001	rd	0000011	LH
imm[11:0]		rs1	010	rd	0000011	LW
imm[11:0]		rs1	100	rd	0000011	LBU
imm[11:0]		rs1	101	rd	0000011	LHU
imm[11:5]	rs2	rs1	000	imm[4:0]	0100011	SB
imm[11:5]	rs2	rs1	001	imm[4:0]	0100011	SH
imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	SW
imm[11:0]		rs1	000	rd	0010011	ADDI
imm[11:0]		rs1	010	rd	0010011	SLTI
imm[11:0]		rs1	011	rd	0010011	SLTIU
imm[11:0]		rs1	100	rd	0010011	XORI
imm[11:0]		rs1	110	rd	0010011	ORI
imm[11:0]		rs1	111	rd	0010011	ANDI
0000000	shamt	rs1	001	rd	0010011	SLLI
0000000	shamt	rs1	101	rd	0010011	SRLI
0100000	shamt	rs1	101	rd	0010011	SRAI
0000000	rs2	rs1	000	rd	0110011	ADD
0100000	rs2	rs1	000	rd	0110011	SUB
0000000	rs2	rs1	001	rd	0110011	SLL
0000000	rs2	rs1	010	rd	0110011	SLT
0000000	rs2	rs1	011	rd	0110011	SLTU
0000000	rs2	rs1	100	rd	0110011	XOR
0000000	rs2	rs1	101	rd	0110011	SRL
0100000	rs2	rs1	101	rd	0110011	SRA
0000000	rs2	rs1	110	rd	0110011	OR
0000000	rs2	rs1	111	rd	0110011	AND
fm | pred | succ		rs1	000	rd	0001111	FENCE
000000000000		00000	000	00000	1110011	ECALL
000000000001		00000	000	00000	1110011	EBREAK

2. RISC-V 指令集

Category Name	Fmt	RV32I Base	
Shifts			
Shift Left Logical	R	SLL	rd,rs1,rs2
Shift Left Log.Imm.	I	SLLI	rd,rs1,shamt
Shift Right Logical	R	SRL	rd,rs1,rs2
Shift Right Log.Imm.	I	SRLI	rd,rs1,shamt
Shift Right Arithmetic	R	SRA	rd,rs1,rs2
Shift Right Arith.Imm.	I	SRAI	rd,rs1,shamt
Arithmetic			
ADD	R	ADD	rd,rs1,rs2
ADD Immediate	I	ADDI	rd,rs1,imm
SUBtract	R	SUB	rd,rs1,rs2
Load Upper Imm	U	LUI	rd,imm
Add Upper Imm to PC	U	AUIPC	rd,imm
Logical			
XOR	R	XOR	rd,rs1,rs2
XOR Immediate	I	XORI	rd,rs1,imm
OR	R	OR	rd,rs1,rs2
OR Immediate	I	ORI	rd,rs1,imm
AND	R	AND	rd,rs1,rs2
AND Immediate	I	ANDI	rd,rs1,imm

测试环境:

部件	配置
CPU	core i7
内存	16GB
操作系统	windows 10

测试结果:

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。