

The background of the page features a large, light gray watermark of the Hunan University seal. The seal is circular, with a traditional Chinese architectural structure in the center, surrounded by a laurel wreath. The text 'HUNAN UNIVERSITY' is written in English around the inner circle, and '湖南大学' is written in Chinese characters around the outer circle. At the bottom of the seal, the founding year '1926' is visible.

湖南大学

HUNAN UNIVERSITY

RISC-V 基本指令集汇编器 设计与实现

班级：智能 1602

学号：201608010723

姓名：施园

一、实验目的

完成一个模拟 RISC-V 的基本整数指令集 RV32 的汇编器设计，实现

汇编指令向二进制的转换。

二、实验要求

- 1、硬件设计采用 VHDL 或 Verilog 语言，软件设计采用 C/C++或 SystemC 语言，其它语言例如 Chisel、MyHDL 等也可选。
- 2、实验报告采用 markdown 语言，或者直接上传 PDF 文档
- 3、实验最终提交所有代码和文档。

三、实验过程

这份工作在暑假的时候没有做过，所以本次实验相对困难一点。汇编器是将汇编语言翻译成机器语言的程序，汇编程序生成目标代码，经过链接器链接后生成可执行代码才可执行。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，汇编器将汇编语言原文件会变成机器可执行的代码。

其中汇编器程序设计框架如下：

<标号>: add x1, x2, x3;

<标号>:10101010101101001;

汇编程序文件 file.asm;

一行一个汇编语句;

初始化地址计数器 addr_counter = 0;

while(file.asm 没有到文件尾) {

读入一行

while(读入的是纯标号且不是文件尾) { 继续读一行 }

拆开行，得到标号（有可能没有），操作码或者伪指令助记符，操作数；

if(有标号) { 记下标号和当前地址计数器的值，保存到符号表；

查看未决汇编语句是否需要这个标号，并解决；

}

if(操作码助记符) {

生成操作码编码；

操作数 -> 寄存器编号或者立即数

if(操作数是标号) {

查找符号表，如果查到，计算得到偏移量；

如果没查到，记下当前汇编语句和地址；

}

生成指令的二进制表示 }

```
49
50 void getnum(string s){
51     memset(d,0,sizeof(d))
52     int j=0;
53     for(int i=0;i<s.length();i++){
54         if(s[i]!='0' && s[i]!='9'){
55             d[j] = d[j]*10+int(s[i]-'0');
56             if(i+1<s.length() && (s[i+1]<'0' || s[i+1]>'9')) j++;
57         }
58     }
59
60     void solveR(int op, int rd, int rs1, int rs2){ // 整数 ADD/SLT/SLTU/AND/OR/XOR/SLL/SRL/SUB/SRA
61     }
62     void solveI(int op, int rd, int rs1, int imm){ //Load,I-type, 整数计算 ADDI/SLTI(U)/ANDI/ORI/XORI/SLLI/SRLI/SRAI JALR
63     cout<< bitset<32>(op | (rd<<7) | (rs1<<15) | (imm<<20)) <<endl;
64     }
65     void solveS(int op, int rs1, int rs2, int imm){ //S-type
66     cout<< bitset<32>(op | (rs2<<20) | (rs1<<15) | ((imm & 0xfe0)<<20) | ((imm & 0x1f)<<7)) <<endl;
67     }
68
69     void solveU(int op, int rd, int imm){ //U-type
70     cout<< bitset<32>(op | (rd<<7) | (imm<<12))<<endl;
71     }
72
73     void solveSB(int op, int rs1, int rs2, int imm){ //S-type和B-type
74     cout<< bitset<32>(op | (rs2<<20) | (rs1<<15) | ((imm & (1<<11))<<20) | ((imm & 0x3f0)<<21) | ((imm & (1<<10))>>3) | ((imm & 0xf)<<
75     }
76
77     void solveUJA(int op, int rd, int imm){
78     cout<< bitset<32>(op | (rd<<7) | ((imm & (1<<19))<<12) | ((imm & 0x3ff)<<21) | ((imm & (1<<10))<<10) | ((imm & 0x7f00)<<1)) <<endl;
79     }
```

对汇编器进行测试(其中操作系统为 windows10)结果如下:

测试输入如下:

- 1、writeWord(0,0x0013ab73);//0000000000001 0011 1010 1011 01110011
- 2、writeWord(4,0x0013db73);//0000000000001 0011 1101 1011 01110011
- 3、writeWord(8,0x0013fb73);//0000000000001 0011 1111 1011 01110011
- 4、writeWord(12,0x0000100f);//0000000000000 0000 0001 0000 00001111
- 5、writeWord(16,0x00100073);//0000000000001 0000 0000 0000 01110011

输入如下:

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

ADDI 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,r1,10

输出如下：

00000000001000001000000110110011

01000000001000001000000110110011

00000000001000001100000110110011

00000000001000001110000110110011

00000000001000001111000110110011

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

四、分析和总结

从测试记录可以看到，汇编器实现了对指令文件的读入，并将汇编指令转换为二进制。综上所述，所编写的汇编器实现了所要求的的功能，完成了实验目标。