浙江大学

本科实验报告

课程名称：	计算机组成与设计
姓　　名：	李依林
院　　系：	信息与电子工程学院
专　　业：	信息工程
学　　号：	3170101212
指导教师：	刘鹏

2019 年 11 月 25 日

专业：信息工程
姓名：李依林
学号：3170101212
日期：2019.11.25
地点：

浙江大学实验报告

课程名称：计算机组成与设计 指导老师：刘鹏 成绩：_____
实验名称：riscv-simulator 实验类型：设计

装 订 线

1. Absract

In this project, I try to simulate the scaled-down version of the RISC-V ISA and implement a 32-bit architecture. In addition, this project mainly consists of two parts including an assembler and a simulator. Since the teacher has given the basic code, my task is to analyze the code, complete it and run to test it.

2. asm: an assembler for the reduced RISC-V ISA

2.1. Function analysis

This function is used to get the label address to constituent immediate number in the case of lw, sw and beq instruction.

```
int get_label_address(char *s)
//char *s;
{
    int i;
    for (i = 0; i < NumValidLabels; i++) {
        if (strlen(Labels[i]) == 0) {
            return -1;
        }
        if (strcmp(Labels[i], s) == 0) {
            return Addresses[i];
        }
    }
    return -1;
}
```

2.2. Task analysis

The main function mainly consists of two parts. The first part is to map symbols to addresses. The second part is to print machine code, with symbols filled in as addresses. And if the input is wrong, the process will stop and report the mistake to you. And since there is an example of "add", I only need to follow the example to complete the rest.

When it comes to the lw, sw, and beq instructions, the imm field can either be a decimal value or a label. So I first decide whether the immediate number consists of a label or a decimal number. Then I use different ways to do it.

For lw and sw instructions, the assembler inserts the absolute address corresponding to the label. So I use the function get_label_address(char *s) to get it. For beq instructions, the assembler computes a PC-relative offset with respect to the label. So I use (get_label_address(arg2)-address) to calculate

the offset.

2.3. My code

```
////////////////////////////////////////////////////////////
        else if (!strcmp(opcode, "sub")) {
            num = (SUB_FUNC7 << FUNCT7_SHIFT) | (atoi(arg2) << RS2_SHIF
T) | (atoi(arg1) << RS1_SHIFT) | SUB_FUNC3<<FUNCT3_SHIFT | (atoi(arg0)
<< RD_SHIFT) | REG_REG_OP;
        }
        else if (!strcmp(opcode, "sll")) {
            num = (SLL_FUNC7 << FUNCT7_SHIFT) | (atoi(arg2) << RS2_SHIF
T) | (atoi(arg1) << RS1_SHIFT) | SLL_FUNC3<<FUNCT3_SHIFT | (atoi(arg0)
<< RD_SHIFT) | REG_REG_OP;
        }
        else if (!strcmp(opcode, "srl")) {
            num = (SRL_FUNC7 << FUNCT7_SHIFT) | (atoi(arg2) << RS2_SHIF
T) | (atoi(arg1) << RS1_SHIFT) | SRL_FUNC3<<FUNCT3_SHIFT | (atoi(arg0)
<< RD_SHIFT) | REG_REG_OP;
        }
        else if (!strcmp(opcode, "or")) {
            num = (OR_FUNC7 << FUNCT7_SHIFT) | (atoi(arg2) << RS2_SHIFT
) | (atoi(arg1) << RS1_SHIFT) | OR_FUNC3<<FUNCT3_SHIFT | (atoi(arg0) <<
 RD_SHIFT) | REG_REG_OP;
        }
        else if (!strcmp(opcode, "and")) {
            num = (AND_FUNC7 << FUNCT7_SHIFT) | (atoi(arg2) << RS2_SHIF
T) | (atoi(arg1) << RS1_SHIFT) | AND_FUNC3<<FUNCT3_SHIFT | (atoi(arg0)
<< RD_SHIFT) | REG_REG_OP;
        }
        else if (!strcmp(opcode, "addi")) {
            num = (atoi(arg2) << ADDI_IMM_SHIFT) | (atoi(arg1) << RS1_S
HIFT) | ADDI_FUNC3<<FUNCT3_SHIFT | (atoi(arg0) << RD_SHIFT) | ADDI_OP;
        }
        else if (!strcmp(opcode, "lw")) {
            int flag;
            flag = get_label_address(arg2);
            if (flag  < 0)
            {
```

装 订 线

```
                num = (atoi(arg2) << LW_IMM_SHIFT) | (atoi(arg1) << RS1
_SHIFT) | LW_FUNC3<<FUNCT3_SHIFT | (atoi(arg0) << RD_SHIFT) | LW_OP;
            }
            else
            {
                num = (get_label_address(arg2) << LW_IMM_SHIFT) | (atoi
(arg1) << RS1_SHIFT) | LW_FUNC3<<FUNCT3_SHIFT | (atoi(arg0) << RD_SHIFT
) | LW_OP;
            }
        }
        else if (!strcmp(opcode, "sw")) {
            int flag;
            flag = get_label_address(arg2);
            if (flag  < 0)
            {
                num = ( (atoi(arg2) & 0xFE0) << 20) | (atoi(arg0) << RS
2_SHIFT) | (atoi(arg1) << RS1_SHIFT) | SW_FUNC3<<FUNCT3_SHIFT | ( (atoi
(arg2) & 0x1F) << RD_SHIFT) | SW_OP;
            }
            else
            {
                num = (((get_label_address(arg2)) & 0xFE0) << 20) | (at
oi(arg0) << RS2_SHIFT) | (atoi(arg1) << RS1_SHIFT) | SW_FUNC3<<FUNCT3_S
HIFT | ((get_label_address(arg2) & 0x1F) << RD_SHIFT) | SW_OP;
            }

        }
         else if (!strcmp(opcode, "beq")) {
            int flag;
            flag = get_label_address(arg2);
            if (flag  < 0)
            {
                num = ( (atoi(arg2) & 0x1000)<< 19) | ( (atoi(arg2) & 0
x7E0) << 20) | (atoi(arg1) << RS2_SHIFT) | (atoi(arg0) << RS1_SHIFT) |
BEQ_FUNC3<<FUNCT3_SHIFT | ( (atoi(arg2) & 0x1E) << RD_SHIFT) | ( (atoi(
arg2) & 0x800) >> 4)| BEQ_OP;
            }
            else
```

装 订 线

```
            {
                num = ( ((get_label_address(arg2)-
address) & 0x1000)<< 19) | ( ((get_label_address(arg2)-
address) & 0x7E0) << 20) | (atoi(arg1) << RS2_SHIFT) | (atoi(arg0) << R
S1_SHIFT) | BEQ_FUNC3<<FUNCT3_SHIFT | ( ((get_label_address(arg2)-
address) & 0x1E) << RD_SHIFT) | ( ((get_label_address(arg2)-
address) & 0x800) >> 4)| BEQ_OP;
            }
        }
 ////////////////////////////////////////////////////////
```

3. Simulator

3.1. Analysis

In this program, we first define a struct variable. Then there are two functions including printState(Pstate) and run(Pstate). The first one is used to show the result, and the other one is used to simulator. And my task is to assign values to some variables in the first part and to define specific operational content in the second part.

3.2. My code

Part1:

```
////////////////////////////////////////////////////////
        rs1 = ((state->mem[word_pc] ) >> 15) & REG_MASK;
        rs2 = ((state->mem[word_pc] ) >> 20) & REG_MASK;
        rd  = ((state->mem[word_pc] ) >> 7 ) & REG_MASK;

        func3 = ((state->mem[word_pc] ) >> 12 ) & FUNC3_MASK;
        func7 = ((state->mem[word_pc] ) >> 25 ) & FUNC7_MASK;
        immField = ((state->mem[word_pc] ) >> 20 ) & ADDI_IMM_MASK;

        imm = ((func7 >> 6) << 11) | ((rd & 0x1) << 10) | ((func7 << 4)
 & 0x3FF) | ((rd >> 1)) ;

        state->pc=state->pc + 4;
////////////////////////////////////////////////////////
```

Part2:

```
////////////////////////////////////////////////////////
        else if (opcode == LW_OP) {
            state->reg[rd] = state->mem[(state->reg[rs1]) + immField>>2
];
        }
        else if (opcode == SW_OP) {
```

装 订 线

```
            state->mem[(state->reg[rs1]) + ((func7 << 5) | rd)>>2] = st
ate->reg[rs2];
        }
        else if (opcode == BEQ_OP) {
            if (state->reg[rs1] == state->reg[rs2])
            state->pc = state->pc + imm-4;
        }
        else if (opcode == REG_REG_OP) {
            if ((func7 == ADD_FUNC7) && (func3 == ADD_FUNC3))
            {
                state->reg[rd] = state->reg[rs1] + state->reg[rs2];
            }
            else if ((func7 == SUB_FUNC7) && (func3 == SUB_FUNC3))
            {
                state->reg[rd] = state->reg[rs1] - state->reg[rs2];
            }
            else if ((func7 == SLL_FUNC7) && (func3 == SLL_FUNC3))
            {
                state->reg[rd] = state->reg[rs1] << state->reg[rs2];
            }
         else if ((func7 == SRL_FUNC7) && (func3 == SRL_FUNC3))
            {
                state->reg[rd] = state->reg[rs1] >> state->reg[rs2];
            }
            else if ((func7 == AND_FUNC7) && (func3 == AND_FUNC3))
            {
                state->reg[rd] = state->reg[rs1] & state->reg[rs2];
            }
            else if ((func7 == OR_FUNC7) && (func3 == OR_FUNC3))
            {
                state->reg[rd] = state->reg[rs1] | state->reg[rs2];
            }
        }
////////////////////////////////////////////////////////////
```

4. Test

4.1. Test1

Code:

```
     lw  21  0   op1 reg[21] <- op1

    lw  22  0   op2 reg[22] <- op2
    lw  23  0   op3 reg[23] <- op3
    add 24  21  22  reg[24] <- reg[21] + reg[22]
    sub 25  24  23  reg[25] <- reg[24] - reg[23]
    sw  25  0   answer  reg[25] -> answer
done    halt
op1 .fill   50
op2 .fill   30
op3 .fill   20
answer  .fill   0
```

Result1:

01c02a83
02002b03
02402b83
016a8c33
417c0cb3
03902423
0000003f
00000032
0000001e
00000014
00000000

The result1 is correct.

Result2:

memory[0]=01c02a83
memory[1]=02002b03
memory[2]=02402b83
memory[3]=016a8c33
memory[4]=417c0cb3
memory[5]=03902423
memory[6]=0000003f
memory[7]=00000032
memory[8]=0000001e
memory[9]=00000014
memory[10]=00000000

state after cycle 0:
    pc=4
    memory:
        mem[0] 0x1c02a83 (29371011)
        mem[1] 0x2002b03 (33565443)

装 订 线

mem[2] 0x2402b83 (37759875)
mem[3] 0x16a8c33 (23759923)
mem[4] 0x417c0cb3 (1098648755)
mem[5] 0x3902423 (59778083)
mem[6] 0x3f (63)
mem[7] 0x32 (50)
mem[8] 0x1e (30)
mem[9] 0x14 (20)
mem[10] 0x0 (0)
registers:
reg[0] 0x0 (0)
reg[1] 0x0 (0)
reg[2] 0x0 (0)
reg[3] 0x0 (0)
reg[4] 0x0 (0)
reg[5] 0x0 (0)
reg[6] 0x0 (0)
reg[7] 0x0 (0)
reg[8] 0x0 (0)
reg[9] 0x0 (0)
reg[10] 0x0 (0)
reg[11] 0x0 (0)
reg[12] 0x0 (0)
reg[13] 0x0 (0)
reg[14] 0x0 (0)
reg[15] 0x0 (0)
reg[16] 0x0 (0)
reg[17] 0x0 (0)
reg[18] 0x0 (0)
reg[19] 0x0 (0)
reg[20] 0x0 (0)
reg[21] 0x32 (50)
reg[22] 0x0 (0)
reg[23] 0x0 (0)
reg[24] 0x0 (0)
reg[25] 0x0 (0)
reg[26] 0x0 (0)
reg[27] 0x0 (0)
reg[28] 0x0 (0)
reg[29] 0x0 (0)
reg[30] 0x0 (0)
reg[31] 0x0 (0)

state after cycle 1:

pc=8
memory:
mem[0] 0x1c02a83 (29371011)
mem[1] 0x2002b03 (33565443)
mem[2] 0x2402b83 (37759875)
mem[3] 0x16a8c33 (23759923)
mem[4] 0x417c0cb3 (1098648755)
mem[5] 0x3902423 (59778083)
mem[6] 0x3f (63)
mem[7] 0x32 (50)
mem[8] 0x1e (30)
mem[9] 0x14 (20)
mem[10] 0x0 (0)
registers:
reg[0] 0x0 (0)
reg[1] 0x0 (0)
reg[2] 0x0 (0)
reg[3] 0x0 (0)
reg[4] 0x0 (0)
reg[5] 0x0 (0)
reg[6] 0x0 (0)
reg[7] 0x0 (0)
reg[8] 0x0 (0)
reg[9] 0x0 (0)
reg[10] 0x0 (0)
reg[11] 0x0 (0)
reg[12] 0x0 (0)
reg[13] 0x0 (0)
reg[14] 0x0 (0)
reg[15] 0x0 (0)
reg[16] 0x0 (0)
reg[17] 0x0 (0)
reg[18] 0x0 (0)
reg[19] 0x0 (0)
reg[20] 0x0 (0)
reg[21] 0x32 (50)
reg[22] 0x1e (30)
reg[23] 0x0 (0)
reg[24] 0x0 (0)
reg[25] 0x0 (0)
reg[26] 0x0 (0)
reg[27] 0x0 (0)
reg[28] 0x0 (0)
reg[29] 0x0 (0)

reg[30] 0x0 (0)
reg[31] 0x0 (0)

state after cycle 2:
pc=12
memory:
mem[0] 0x1c02a83 (29371011)
mem[1] 0x2002b03 (33565443)
mem[2] 0x2402b83 (37759875)
mem[3] 0x16a8c33 (23759923)
mem[4] 0x417c0cb3 (1098648755)
mem[5] 0x3902423 (59778083)
mem[6] 0x3f (63)
mem[7] 0x32 (50)
mem[8] 0x1e (30)
mem[9] 0x14 (20)
mem[10] 0x0 (0)
registers:
reg[0] 0x0 (0)
reg[1] 0x0 (0)
reg[2] 0x0 (0)
reg[3] 0x0 (0)
reg[4] 0x0 (0)
reg[5] 0x0 (0)
reg[6] 0x0 (0)
reg[7] 0x0 (0)
reg[8] 0x0 (0)
reg[9] 0x0 (0)
reg[10] 0x0 (0)
reg[11] 0x0 (0)
reg[12] 0x0 (0)
reg[13] 0x0 (0)
reg[14] 0x0 (0)
reg[15] 0x0 (0)
reg[16] 0x0 (0)
reg[17] 0x0 (0)
reg[18] 0x0 (0)
reg[19] 0x0 (0)
reg[20] 0x0 (0)
reg[21] 0x32 (50)
reg[22] 0x1e (30)
reg[23] 0x14 (20)
reg[24] 0x0 (0)
reg[25] 0x0 (0)

装 订 线

reg[26] 0x0 (0)
reg[27] 0x0 (0)
reg[28] 0x0 (0)
reg[29] 0x0 (0)
reg[30] 0x0 (0)
reg[31] 0x0 (0)

state after cycle 3:
pc=16
memory:
mem[0] 0x1c02a83 (29371011)
mem[1] 0x2002b03 (33565443)
mem[2] 0x2402b83 (37759875)
mem[3] 0x16a8c33 (23759923)
mem[4] 0x417c0cb3 (1098648755)
mem[5] 0x3902423 (59778083)
mem[6] 0x3f (63)
mem[7] 0x32 (50)
mem[8] 0x1e (30)
mem[9] 0x14 (20)
mem[10] 0x0 (0)
registers:
reg[0] 0x0 (0)
reg[1] 0x0 (0)
reg[2] 0x0 (0)
reg[3] 0x0 (0)
reg[4] 0x0 (0)
reg[5] 0x0 (0)
reg[6] 0x0 (0)
reg[7] 0x0 (0)
reg[8] 0x0 (0)
reg[9] 0x0 (0)
reg[10] 0x0 (0)
reg[11] 0x0 (0)
reg[12] 0x0 (0)
reg[13] 0x0 (0)
reg[14] 0x0 (0)
reg[15] 0x0 (0)
reg[16] 0x0 (0)
reg[17] 0x0 (0)
reg[18] 0x0 (0)
reg[19] 0x0 (0)
reg[20] 0x0 (0)
reg[21] 0x32 (50)

装 订 线

```
        reg[22] 0x1e   (30)
        reg[23] 0x14   (20)
        reg[24] 0x50   (80)
        reg[25] 0x0    (0)
        reg[26] 0x0    (0)
        reg[27] 0x0    (0)
        reg[28] 0x0    (0)
        reg[29] 0x0    (0)
        reg[30] 0x0    (0)
        reg[31] 0x0    (0)

state after cycle 4:
    pc=20
    memory:
        mem[0] 0x1c02a83 (29371011)
        mem[1] 0x2002b03 (33565443)
        mem[2] 0x2402b83 (37759875)
        mem[3] 0x16a8c33 (23759923)
        mem[4] 0x417c0cb3      (1098648755)
        mem[5] 0x3902423 (59778083)
        mem[6] 0x3f   (63)
        mem[7] 0x32   (50)
        mem[8] 0x1e   (30)
        mem[9] 0x14   (20)
        mem[10] 0x0   (0)
    registers:
        reg[0] 0x0     (0)
        reg[1] 0x0     (0)
        reg[2] 0x0     (0)
        reg[3] 0x0     (0)
        reg[4] 0x0     (0)
        reg[5] 0x0     (0)
        reg[6] 0x0     (0)
        reg[7] 0x0     (0)
        reg[8] 0x0     (0)
        reg[9] 0x0     (0)
        reg[10] 0x0    (0)
        reg[11] 0x0    (0)
        reg[12] 0x0    (0)
        reg[13] 0x0    (0)
        reg[14] 0x0    (0)
        reg[15] 0x0    (0)
        reg[16] 0x0    (0)
        reg[17] 0x0    (0)
```

装 订 线

```
        reg[18] 0x0     (0)
        reg[19] 0x0     (0)
        reg[20] 0x0     (0)
        reg[21] 0x32    (50)
        reg[22] 0x1e    (30)
        reg[23] 0x14    (20)
        reg[24] 0x50    (80)
        reg[25] 0x3c    (60)
        reg[26] 0x0     (0)
        reg[27] 0x0     (0)
        reg[28] 0x0     (0)
        reg[29] 0x0     (0)
        reg[30] 0x0     (0)
        reg[31] 0x0     (0)

state after cycle 5:
    pc=24
    memory:
        mem[0] 0x1c02a83 (29371011)
        mem[1] 0x2002b03 (33565443)
        mem[2] 0x2402b83 (37759875)
        mem[3] 0x16a8c33 (23759923)
        mem[4] 0x417c0cb3       (1098648755)
        mem[5] 0x3902423 (59778083)
        mem[6] 0x3f   (63)
        mem[7] 0x32   (50)
        mem[8] 0x1e   (30)
        mem[9] 0x14   (20)
        mem[10] 0x3c (60)
    registers:
        reg[0] 0x0      (0)
        reg[1] 0x0      (0)
        reg[2] 0x0      (0)
        reg[3] 0x0      (0)
        reg[4] 0x0      (0)
        reg[5] 0x0      (0)
        reg[6] 0x0      (0)
        reg[7] 0x0      (0)
        reg[8] 0x0      (0)
        reg[9] 0x0      (0)
        reg[10] 0x0     (0)
        reg[11] 0x0     (0)
        reg[12] 0x0     (0)
        reg[13] 0x0     (0)
```

装

订

线

```
        reg[14] 0x0     (0)
        reg[15] 0x0     (0)
        reg[16] 0x0     (0)
        reg[17] 0x0     (0)
        reg[18] 0x0     (0)
        reg[19] 0x0     (0)
        reg[20] 0x0     (0)
        reg[21] 0x32    (50)
        reg[22] 0x1e    (30)
        reg[23] 0x14    (20)
        reg[24] 0x50    (80)
        reg[25] 0x3c    (60)
        reg[26] 0x0     (0)
        reg[27] 0x0     (0)
        reg[28] 0x0     (0)
        reg[29] 0x0     (0)
        reg[30] 0x0     (0)
        reg[31] 0x0     (0)

machine halted
total of 7 instructions executed
state after cycle 6:
    pc=28
    memory:
        mem[0] 0x1c02a83 (29371011)
        mem[1] 0x2002b03 (33565443)
        mem[2] 0x2402b83 (37759875)
        mem[3] 0x16a8c33 (23759923)
        mem[4] 0x417c0cb3       (1098648755)
        mem[5] 0x3902423 (59778083)
        mem[6] 0x3f   (63)
        mem[7] 0x32   (50)
        mem[8] 0x1e   (30)
        mem[9] 0x14   (20)
        mem[10] 0x3c (60)
    registers:
        reg[0] 0x0      (0)
        reg[1] 0x0      (0)
        reg[2] 0x0      (0)
        reg[3] 0x0      (0)
        reg[4] 0x0      (0)
        reg[5] 0x0      (0)
        reg[6] 0x0      (0)
        reg[7] 0x0      (0)
```

reg[8] 0x0 (0)
reg[9] 0x0 (0)
reg[10] 0x0 (0)
reg[11] 0x0 (0)
reg[12] 0x0 (0)
reg[13] 0x0 (0)
reg[14] 0x0 (0)
reg[15] 0x0 (0)
reg[16] 0x0 (0)
reg[17] 0x0 (0)
reg[18] 0x0 (0)
reg[19] 0x0 (0)
reg[20] 0x0 (0)
reg[21] 0x32 (50)
reg[22] 0x1e (30)
reg[23] 0x14 (20)
reg[24] 0x50 (80)
reg[25] 0x3c (60)
reg[26] 0x0 (0)
reg[27] 0x0 (0)
reg[28] 0x0 (0)
reg[29] 0x0 (0)
reg[30] 0x0 (0)
reg[31] 0x0 (0)

The result2 is correct.

4.2. Test2

Code:

```
   lw  21  0   36  reg[21] <- op1

    lw  22  0   op1 reg[22] <- op1
    beq 21  22  lab if(reg[21] == reg[22]) then lab
    lw  21  0   op3 reg[21] <- op3
lab add 24  21  22  reg[24] <- reg[21] + reg[22]
    sub 25  24  23  reg[25] <- reg[24] - reg[23]
    sw  25  0   answer  reg[25] -> answer
    sw  23  0   48  reg[23] -> M[12]
done    halt
op1 .fill   50
op2 .fill   30
op3 .fill   20
answer  .fill   0
```

Result1：

02402a83
02402b03
016a8463
02c02a83
016a8c33
417c0cb3
03902823
03702823
0000003f
00000032
0000001e
00000014
00000000
The result1 is correct.
Result2:
memory[0]=02402a83
memory[1]=02402b03
memory[2]=016a8463
memory[3]=02c02a83
memory[4]=016a8c33
memory[5]=417c0cb3
memory[6]=03902823
memory[7]=03702823
memory[8]=0000003f
memory[9]=00000032
memory[10]=0000001e
memory[11]=00000014
memory[12]=00000000

state after cycle 0:
pc=4
memory:
mem[0] 0x2402a83 (37759619)
mem[1] 0x2402b03 (37759747)
mem[2] 0x16a8463 (23757923)
mem[3] 0x2c02a83 (46148227)
mem[4] 0x16a8c33 (23759923)
mem[5] 0x417c0cb3 (1098648755)
mem[6] 0x3902823 (59779107)
mem[7] 0x3702823 (57681955)
mem[8] 0x3f (63)
mem[9] 0x32 (50)
mem[10] 0x1e (30)
mem[11] 0x14 (20)

装 订 线

装 订 线

```
        mem[12] 0x0  (0)
    registers:
        reg[0] 0x0      (0)
        reg[1] 0x0      (0)
        reg[2] 0x0      (0)
        reg[3] 0x0      (0)
        reg[4] 0x0      (0)
        reg[5] 0x0      (0)
        reg[6] 0x0      (0)
        reg[7] 0x0      (0)
        reg[8] 0x0      (0)
        reg[9] 0x0      (0)
        reg[10] 0x0     (0)
        reg[11] 0x0     (0)
        reg[12] 0x0     (0)
        reg[13] 0x0     (0)
        reg[14] 0x0     (0)
        reg[15] 0x0     (0)
        reg[16] 0x0     (0)
        reg[17] 0x0     (0)
        reg[18] 0x0     (0)
        reg[19] 0x0     (0)
        reg[20] 0x0     (0)
        reg[21] 0x32    (50)
        reg[22] 0x0     (0)
        reg[23] 0x0     (0)
        reg[24] 0x0     (0)
        reg[25] 0x0     (0)
        reg[26] 0x0     (0)
        reg[27] 0x0     (0)
        reg[28] 0x0     (0)
        reg[29] 0x0     (0)
        reg[30] 0x0     (0)
        reg[31] 0x0     (0)

state after cycle 1:
    pc=8
    memory:
        mem[0] 0x2402a83 (37759619)
        mem[1] 0x2402b03 (37759747)
        mem[2] 0x16a8463 (23757923)
        mem[3] 0x2c02a83 (46148227)
        mem[4] 0x16a8c33 (23759923)
        mem[5] 0x417c0cb3       (1098648755)
```

装 订 线

mem[6] 0x3902823 (59779107)
mem[7] 0x3702823 (57681955)
mem[8] 0x3f (63)
mem[9] 0x32 (50)
mem[10] 0x1e (30)
mem[11] 0x14 (20)
mem[12] 0x0 (0)
registers:
reg[0] 0x0 (0)
reg[1] 0x0 (0)
reg[2] 0x0 (0)
reg[3] 0x0 (0)
reg[4] 0x0 (0)
reg[5] 0x0 (0)
reg[6] 0x0 (0)
reg[7] 0x0 (0)
reg[8] 0x0 (0)
reg[9] 0x0 (0)
reg[10] 0x0 (0)
reg[11] 0x0 (0)
reg[12] 0x0 (0)
reg[13] 0x0 (0)
reg[14] 0x0 (0)
reg[15] 0x0 (0)
reg[16] 0x0 (0)
reg[17] 0x0 (0)
reg[18] 0x0 (0)
reg[19] 0x0 (0)
reg[20] 0x0 (0)
reg[21] 0x32 (50)
reg[22] 0x32 (50)
reg[23] 0x0 (0)
reg[24] 0x0 (0)
reg[25] 0x0 (0)
reg[26] 0x0 (0)
reg[27] 0x0 (0)
reg[28] 0x0 (0)
reg[29] 0x0 (0)
reg[30] 0x0 (0)
reg[31] 0x0 (0)

state after cycle 2:
pc=16
memory:

mem[0] 0x2402a83 (37759619)
mem[1] 0x2402b03 (37759747)
mem[2] 0x16a8463 (23757923)
mem[3] 0x2c02a83 (46148227)
mem[4] 0x16a8c33 (23759923)
mem[5] 0x417c0cb3 (1098648755)
mem[6] 0x3902823 (59779107)
mem[7] 0x3702823 (57681955)
mem[8] 0x3f (63)
mem[9] 0x32 (50)
mem[10] 0x1e (30)
mem[11] 0x14 (20)
mem[12] 0x0 (0)
registers:
reg[0] 0x0 (0)
reg[1] 0x0 (0)
reg[2] 0x0 (0)
reg[3] 0x0 (0)
reg[4] 0x0 (0)
reg[5] 0x0 (0)
reg[6] 0x0 (0)
reg[7] 0x0 (0)
reg[8] 0x0 (0)
reg[9] 0x0 (0)
reg[10] 0x0 (0)
reg[11] 0x0 (0)
reg[12] 0x0 (0)
reg[13] 0x0 (0)
reg[14] 0x0 (0)
reg[15] 0x0 (0)
reg[16] 0x0 (0)
reg[17] 0x0 (0)
reg[18] 0x0 (0)
reg[19] 0x0 (0)
reg[20] 0x0 (0)
reg[21] 0x32 (50)
reg[22] 0x32 (50)
reg[23] 0x0 (0)
reg[24] 0x0 (0)
reg[25] 0x0 (0)
reg[26] 0x0 (0)
reg[27] 0x0 (0)
reg[28] 0x0 (0)
reg[29] 0x0 (0)

```
        reg[30] 0x0     (0)
        reg[31] 0x0     (0)

state after cycle 3:
    pc=20
    memory:
        mem[0] 0x2402a83 (37759619)
        mem[1] 0x2402b03 (37759747)
        mem[2] 0x16a8463 (23757923)
        mem[3] 0x2c02a83 (46148227)
        mem[4] 0x16a8c33 (23759923)
        mem[5] 0x417c0cb3       (1098648755)
        mem[6] 0x3902823 (59779107)
        mem[7] 0x3702823 (57681955)
        mem[8] 0x3f   (63)
        mem[9] 0x32   (50)
        mem[10] 0x1e (30)
        mem[11] 0x14 (20)
        mem[12] 0x0   (0)
    registers:
        reg[0] 0x0      (0)
        reg[1] 0x0      (0)
        reg[2] 0x0      (0)
        reg[3] 0x0      (0)
        reg[4] 0x0      (0)
        reg[5] 0x0      (0)
        reg[6] 0x0      (0)
        reg[7] 0x0      (0)
        reg[8] 0x0      (0)
        reg[9] 0x0      (0)
        reg[10] 0x0     (0)
        reg[11] 0x0     (0)
        reg[12] 0x0     (0)
        reg[13] 0x0     (0)
        reg[14] 0x0     (0)
        reg[15] 0x0     (0)
        reg[16] 0x0     (0)
        reg[17] 0x0     (0)
        reg[18] 0x0     (0)
        reg[19] 0x0     (0)
        reg[20] 0x0     (0)
        reg[21] 0x32    (50)
        reg[22] 0x32    (50)
        reg[23] 0x0     (0)
```

装 订 线

装 订 线

```
        reg[24] 0x64    (100)
        reg[25] 0x0     (0)
        reg[26] 0x0     (0)
        reg[27] 0x0     (0)
        reg[28] 0x0     (0)
        reg[29] 0x0     (0)
        reg[30] 0x0     (0)
        reg[31] 0x0     (0)

state after cycle 4:
    pc=24
    memory:
        mem[0] 0x2402a83 (37759619)
        mem[1] 0x2402b03 (37759747)
        mem[2] 0x16a8463 (23757923)
        mem[3] 0x2c02a83 (46148227)
        mem[4] 0x16a8c33 (23759923)
        mem[5] 0x417c0cb3       (1098648755)
        mem[6] 0x3902823 (59779107)
        mem[7] 0x3702823 (57681955)
        mem[8] 0x3f   (63)
        mem[9] 0x32   (50)
        mem[10] 0x1e (30)
        mem[11] 0x14 (20)
        mem[12] 0x0   (0)
    registers:
        reg[0] 0x0      (0)
        reg[1] 0x0      (0)
        reg[2] 0x0      (0)
        reg[3] 0x0      (0)
        reg[4] 0x0      (0)
        reg[5] 0x0      (0)
        reg[6] 0x0      (0)
        reg[7] 0x0      (0)
        reg[8] 0x0      (0)
        reg[9] 0x0      (0)
        reg[10] 0x0     (0)
        reg[11] 0x0     (0)
        reg[12] 0x0     (0)
        reg[13] 0x0     (0)
        reg[14] 0x0     (0)
        reg[15] 0x0     (0)
        reg[16] 0x0     (0)
        reg[17] 0x0     (0)
```

装 订 线

```
        reg[18] 0x0     (0)
        reg[19] 0x0     (0)
        reg[20] 0x0     (0)
        reg[21] 0x32    (50)
        reg[22] 0x32    (50)
        reg[23] 0x0     (0)
        reg[24] 0x64    (100)
        reg[25] 0x64    (100)
        reg[26] 0x0     (0)
        reg[27] 0x0     (0)
        reg[28] 0x0     (0)
        reg[29] 0x0     (0)
        reg[30] 0x0     (0)
        reg[31] 0x0     (0)

state after cycle 5:
    pc=28
    memory:
        mem[0] 0x2402a83 (37759619)
        mem[1] 0x2402b03 (37759747)
        mem[2] 0x16a8463 (23757923)
        mem[3] 0x2c02a83 (46148227)
        mem[4] 0x16a8c33 (23759923)
        mem[5] 0x417c0cb3       (1098648755)
        mem[6] 0x3902823 (59779107)
        mem[7] 0x3702823 (57681955)
        mem[8] 0x3f   (63)
        mem[9] 0x32   (50)
        mem[10] 0x1e (30)
        mem[11] 0x14 (20)
        mem[12] 0x64 (100)
    registers:
        reg[0] 0x0      (0)
        reg[1] 0x0      (0)
        reg[2] 0x0      (0)
        reg[3] 0x0      (0)
        reg[4] 0x0      (0)
        reg[5] 0x0      (0)
        reg[6] 0x0      (0)
        reg[7] 0x0      (0)
        reg[8] 0x0      (0)
        reg[9] 0x0      (0)
        reg[10] 0x0     (0)
        reg[11] 0x0     (0)
```

装 订 线

reg[12] 0x0 (0)
reg[13] 0x0 (0)
reg[14] 0x0 (0)
reg[15] 0x0 (0)
reg[16] 0x0 (0)
reg[17] 0x0 (0)
reg[18] 0x0 (0)
reg[19] 0x0 (0)
reg[20] 0x0 (0)
reg[21] 0x32 (50)
reg[22] 0x32 (50)
reg[23] 0x0 (0)
reg[24] 0x64 (100)
reg[25] 0x64 (100)
reg[26] 0x0 (0)
reg[27] 0x0 (0)
reg[28] 0x0 (0)
reg[29] 0x0 (0)
reg[30] 0x0 (0)
reg[31] 0x0 (0)

state after cycle 6:
pc=32
memory:
mem[0] 0x2402a83 (37759619)
mem[1] 0x2402b03 (37759747)
mem[2] 0x16a8463 (23757923)
mem[3] 0x2c02a83 (46148227)
mem[4] 0x16a8c33 (23759923)
mem[5] 0x417c0cb3 (1098648755)
mem[6] 0x3902823 (59779107)
mem[7] 0x3702823 (57681955)
mem[8] 0x3f (63)
mem[9] 0x32 (50)
mem[10] 0x1e (30)
mem[11] 0x14 (20)
mem[12] 0x0 (0)
registers:
reg[0] 0x0 (0)
reg[1] 0x0 (0)
reg[2] 0x0 (0)
reg[3] 0x0 (0)
reg[4] 0x0 (0)
reg[5] 0x0 (0)

装 订 线

```
        reg[6] 0x0      (0)
        reg[7] 0x0      (0)
        reg[8] 0x0      (0)
        reg[9] 0x0      (0)
        reg[10] 0x0     (0)
        reg[11] 0x0     (0)
        reg[12] 0x0     (0)
        reg[13] 0x0     (0)
        reg[14] 0x0     (0)
        reg[15] 0x0     (0)
        reg[16] 0x0     (0)
        reg[17] 0x0     (0)
        reg[18] 0x0     (0)
        reg[19] 0x0     (0)
        reg[20] 0x0     (0)
        reg[21] 0x32    (50)
        reg[22] 0x32    (50)
        reg[23] 0x0     (0)
        reg[24] 0x64    (100)
        reg[25] 0x64    (100)
        reg[26] 0x0     (0)
        reg[27] 0x0     (0)
        reg[28] 0x0     (0)
        reg[29] 0x0     (0)
        reg[30] 0x0     (0)
        reg[31] 0x0     (0)

machine halted
total of 8 instructions executed
state after cycle 7:
    pc=36
    memory:
        mem[0] 0x2402a83 (37759619)
        mem[1] 0x2402b03 (37759747)
        mem[2] 0x16a8463 (23757923)
        mem[3] 0x2c02a83 (46148227)
        mem[4] 0x16a8c33 (23759923)
        mem[5] 0x417c0cb3       (1098648755)
        mem[6] 0x3902823 (59779107)
        mem[7] 0x3702823 (57681955)
        mem[8] 0x3f   (63)
        mem[9] 0x32   (50)
        mem[10] 0x1e (30)
        mem[11] 0x14 (20)
```

mem[12] 0x0 (0)
registers:
reg[0] 0x0 (0)
reg[1] 0x0 (0)
reg[2] 0x0 (0)
reg[3] 0x0 (0)
reg[4] 0x0 (0)
reg[5] 0x0 (0)
reg[6] 0x0 (0)
reg[7] 0x0 (0)
reg[8] 0x0 (0)
reg[9] 0x0 (0)
reg[10] 0x0 (0)
reg[11] 0x0 (0)
reg[12] 0x0 (0)
reg[13] 0x0 (0)
reg[14] 0x0 (0)
reg[15] 0x0 (0)
reg[16] 0x0 (0)
reg[17] 0x0 (0)
reg[18] 0x0 (0)
reg[19] 0x0 (0)
reg[20] 0x0 (0)
reg[21] 0x32 (50)
reg[22] 0x32 (50)
reg[23] 0x0 (0)
reg[24] 0x64 (100)
reg[25] 0x64 (100)
reg[26] 0x0 (0)
reg[27] 0x0 (0)
reg[28] 0x0 (0)
reg[29] 0x0 (0)
reg[30] 0x0 (0)
reg[31] 0x0 (0)

The result2 is correct.

5. Comment

This project is a little difficult for me. I spend a lot of time to read the code to fully understand how the simulator works. And finally thanks to the discussion with my friends, I finally made it. And it will also help me to learn the course better.