1. Choose the best answer (22%)

1.1The instructions and data in von Neumann computers are stored in binary form in memory, and the CPU distinguishes them based on:

A、Addressing Mode of data and instructions

B、The storage unit in which instructions and data reside

C、The different phases of the instruction cycle

D、Decoding result of instruction opcode

1.2 64-bit microcomputer indicates the CPU used in the computer

A、has 64-bit registers

B、the number of the registers is 64

C、can handle 64-bit binary numbers at the same time

D、can handle 64-bit characters

1.3 The main frequency of a computer is 1.2GHZ, and it has four kinds of instructions. As shown in the following table, there are the proportion in the benchmark program and CPI.

Instruction Type	Proportion	CPI	Instruction Type	Proportion	CPI
A	50%	2	C	10%	4
B	20%	3	D	20%	5

The MIPS of the CPU is

A、100

B、200

C、400

D、600

解：基准程序 CPI = 2*0.5+3*0.2+4*0.1+5*0.2=3。计算机主频 1.2GHZ=1200MHZ，MIPS=1200/3=400

1.4 Who is the smallest number in the following

A、$(101001)_2$

B、$(101001)_{BCD}$

C、$(52)_8$

D、$(233)_{16}$

1.5 A decimal number is -66, and now it is stored in a 8-bits register. The content of the register is ( ) in hexadecimal

A、C2H

B、BEH

C、BDH

D、42H

解：-66 原码：11000010，补码 10111110=BEH

1.6 The smallest integer that can be represented by an 8-bit two's complement code with three ones and five zeros is

A、-126

B、-125

C、-32

D、-3

解：10000011＝-125

1.7 Two floating-point numbers of the same length but in different formats. Assume that the former has a long exponent and a short mantissa (the fraction), and the latter has a short exponent and a long mantissa. With all other rules being the same, they can represent the range and precision of the number

A、Both have the same range and precision

B、The former can be represented in a large range but with low precision

C、The former can be represented in a large range and with high precision

D、The latter can be represented in a large range and with high precision

1.8 ALU is the core component of the computing unit, and it

A、is sequential logic circuit

B、is combinational logic circuit

C、can only have the functions of number operations

D、cannot have the functions of logic operations

1.9 Assuming that the address code in the instruction gives the effective address of the operand, the instruction takes

A、direct addressing

B、immediate addressing

C、register addressing

D、PC-relative addressing

1.10 The whole function of the controller is

A、decoding the instruction opcode

B、generating a time series signal

C、fetching instructions from main memory、analyzing and generating all relative control signal

D、processing interrupt

1.11 The machine number is 8 bits long and contains one sign bit, BAH is the original code, after shifting left arithmetic in 1-bit and shifting right arithmetic in 1-bit, and the result is

A、74H,EDH

B、74H,DDH

C、F4H,9DH

D、B5H,EDH

解：原码 BAH=1011 1010B，算术左移一位 0111 0100；右移一位 1101 1101

2. Filling the gap(8%)

2.1 The 16-bit complement code A = 0X8FA0, after extending to 32-bit it should be____0Xffff 8fa0_____

2.2 The word length of a computer is 32 bits, and it is addressed in byte and use little endian to store data. Now there is a double variable A, A = 1122_3344_5566_7788H, and it is stored in continuous address at the beginning of 0000_8040H. So what is stored in 0000_8046H is____22H_____

2.3 The float data is usually represented in the IEEE754 single-precision floating-point format. Now the compiler allocates the float variable x in a 32-bit floating-point register RF1, and x = -8.25, so using hexadecimal x should be___C1040000H_____

解：x=-8.25=-1.00001*2³ E=130=10000010

X=1_10000010_00001000000000000000000

2.4 Suppose the program counter (PC) is set to 0x60000000. What range of addresses can be reached using the RISC-V branch if equal (beq) instruction? (In other words, what is the set of possible values for the PC after the branch instruction executes?)

__[0x5FFFF000, 0x60000ffe]_____

3.(15%)Assemble:To convert the RISCV instructions into machine code.

Address(Hex)	RISCV Assembly Instruction	Machine Code(Hex)
600000	Loop: jal x0, L1	0x0700006F
600004	add s2, s3, s4	0x01498933
600008	beq t3, t4, Loop	0xFFDE0CE3
600070	L1: ……	

4. (15%) 、To convert the pseudoinstruction(left) into the shortest sequence of RISCV instructions.

Pseudoinstruction	Function	RISCV instructions
Neg rd,rs	Two's complement	Sub rd,x0,rs
Sltz rd,rs	Rd=(rs<0)?1:0	Slt rd,rs,x0
Mv rd, rs	rd = rs copy the register	addi rd,rs,0; /or rd,rs,x0 /add rd rs x0
Jal offset	Jump and link	Jal x1,offset
Ret	Return from subroutine	Jalr x0,x1,0

5.1 (10%) Translate the following C code to RISC-V assembly code. Use a minimum number of instructions. Assume that the values of A, B, m, and n are in registers x5, x6, x7, and x29, respectively. Also, assume that register x10 holds the base address of the array Q:

```
for(m=0; m<A; m++)
    for(n=0; n<B; n++)
        Q[4*n] = m + n;}
```

The C code can be implemented in RISC-V assembly as follows.

```
LOOPI:
        addi x7, x0, 0       // Init m = 0
        bge x7, x5, ENDI     // While m < A
        addi x30, x10, 0     // x30 = &Q
        addi x29, x0, 0      // Init n = 0
LOOPJ:
        bge x29, x6, ENDJ    // While n < B
        add x31, x7, x29     // x31 = m+n
        sd x31, 0(x30)       // Q[4*n] = x31
        addi x30, x30, 32 // x30 = &Q[4*(n+1)]
        addi x29, x29, 1  // n++
        jal x0, LOOPJ
ENDJ:
        addi x7, x7, 1       // m++;
        jal x0, LOOPI
ENDI:
```

5.2 (10%) For the RISC-V assembly instructions below, what is the corresponding C statement? Assume that the variables a, b, h, m, and n are assigned to registers x5, x6, x7, x28, and x29, respectively. Assume that the base address of the arrays A and B are in registers x10 and x11, respectively.:

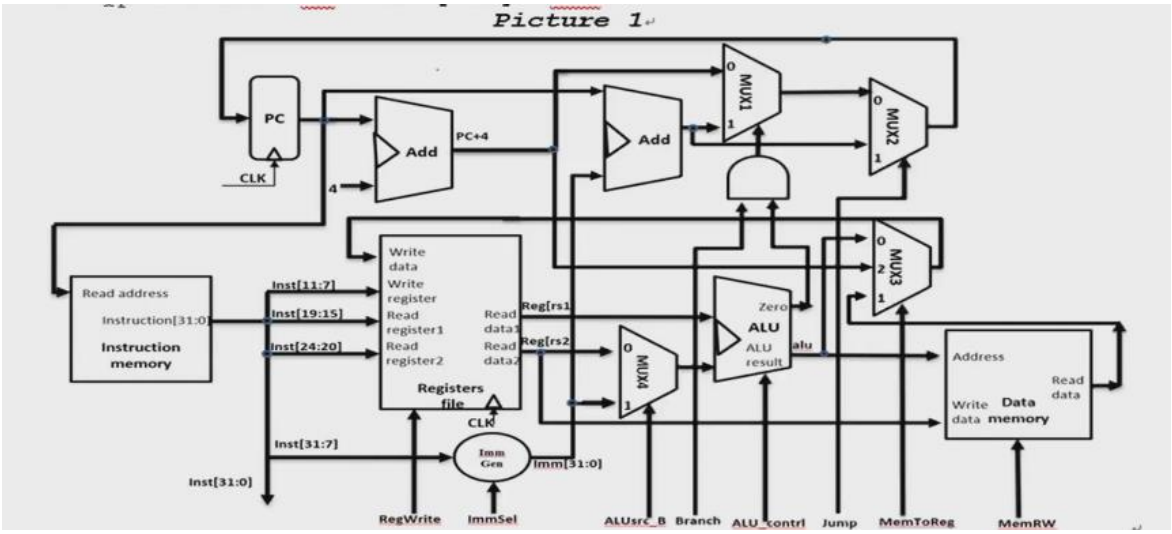
```
slli x30, x5, 3
add x30, x10, x30
slli x31, x6, 3
add x31, x11, x31
ld x5, 0(x30)

addi x12, x30, 8
ld x30, 0(x12)
add x30, x30, x5
sd x30, 0(x31)
```

```
B[b]= A[a] + A[a+1]
slli x30, x5, 3          // x30 = a*8
add x30, x10, x30        // x30 = &A[a]
slli x31, x6, 3          // x31 = b*8
add x31, x11, x31        // x31 = &B[b]
ld x5, 0(x30)            // a = A[a]
addi x12, x30, 8         // x12 = &A[a]+8 (i.e. &A[a+1])
ld x30, 0(x12)           // x30 = A[a+1]
add x30, x30, x5         // x30 = A[a+1] + A[a]
```

```
sd x30, 0(x31)           // B[b] = x30 (i.e. A[a+1] + A[a])
```

6. (20%)Examine the difficulty of adding a proposed **mw rd, rs1,imm** instruction to RISC-V.
Interpretation: **rd = Mem[rs1] +imm**

Picture 1



1) (5%) Which new functional blocks(if any) do we need for this instruction?

additional mux and adder

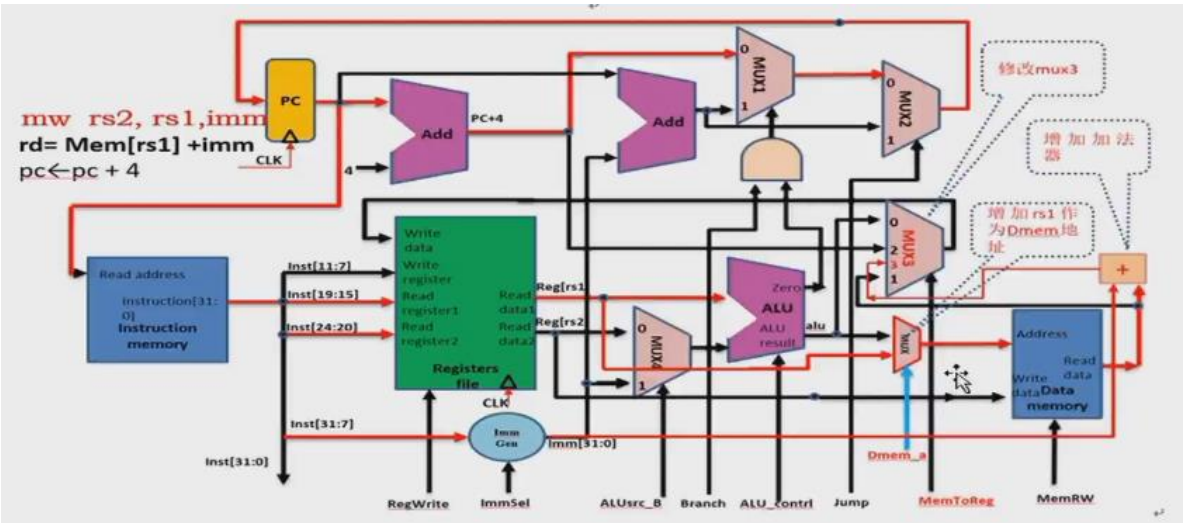
2) (5%) Which existing functional blocks(if any) require modification?

MUX3 need to be modified

3) (6%) Design datapath: Modify the picture1 to demonstrate an implementation of this new instruction

4) (4%) Design control: to set new signals to support this instruction;

参考: **Dmem a**



https://classroom.zju.edu.cn/livingpage?course_id=49844&sub_id=857587&tenant_code=112