# 1 midterm exam

md计组-2024-05-15

1. Choose the best answer (20%)

1.1 What is wrong with the following description of a von Neumann structured computer is:

A、The functions of the program are implemented through the central processor

B、Both instructions and data are represented in binary, and there is no difference in form

C、Instructions are accessed by address, and the data is given directly in the instruction

D、Instructions and data need to be stored in memory before the program can be executed

1.2 The smallest unit of time in a computer is

A、instruction cycle

B、clock cycle

C、CPU cycle

D、execution cycle

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、$(101001)_{BCD}$

B、$(101001)_{2}$

C、$(52)_{8}$

D、$(23)_{16}$

1.5 Assuming signed integers are represented in complement, if the machines of the integer variables x and y are FFFFFFDFH and 00000041H, then the value of x、y and the machines code of x-y are, respectively

A、x=-65,y=41,x-y overflow

B、x=-33,y=65,x-y=FFFFFF9EH

C、x=-65,y=41,x-y=FFFFFF9EH

D、x=-33,y=65,x-y=FFFFFF9DH

解：X 补码：1111 1111 1111 1111 1111 1111 1101 1111，
原码：1000 0000 0000 0000 0000 0000 0010 0001=-33
Y 补码=原码：41H=65;
X-Y=-98 其 8 位补码：9EH,扩展 32 位：FFFFFF9EH

1.6 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

1.7 In the parts given below, whose bit width must be the same as the machine word length is ( )

I. ALU II. Instruction Registers

III. General Registers IV. Floating-point Registers.

A、I、II

B、I、III

C、II、III

D、II、III、IV

mechine word length：32位机，64位机（也就是指支持的数据位宽）

（ALU必须一样，比如64位机的ALU必须支持64位运算；instruction register可以不同，比如课程中就是32位的；general register必须一样，这是拿来做运算的寄存器；floating-point register不用一样，floating-point就是32位或64位的，即使在8位机中也是32单精度，64双精度）

1.8 Operands used in computer instructions can generally be obtained from

A、General register

B、Memory storage unit

C、A register in a peripheral interface

D、The above all

1.9 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

1.10 The machine number X = 11011000, after shifting right logic in 1-bit and shifting right arithmetic in 1-bit, and the result is

A、ECH,6CH

B、DCH,6DH

C、6CH,ECH

D、6CH,EDH

解：原码 X =1101 1000B，逻辑右移一位 0110 1100；算术右移一位 1110 1100

算术移位需要符号扩展

**2. Filling the gap(20%)**

2.1 The 16-bit complement code A = 0X8AF0, after extending to 32-bit it should be ____0Xffff 8af0____

2.2 The word length of a computer is 32 bits, and it is addressed in byte and use big 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____77H____

2.3 The smallest integer that can be represented by an 8-bit two's complement code with three ones and five zeros is -125

解：10000011 = -125

2.2答案存疑：第一个word是1122_3344H还是5566_7788H?

（5566_7788？查一下？）

=>（期末考前问了） double word 第一个word就是低的word

2.4 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

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

__[0x5ff00000, 0x600ffffe]___________

2.6 The word length of a computer is 8 bits, and the type of variables x, y, and z is integer (represented by complement).

X = 11110100, Y = 10110000, if Z = X/2 + 2Y, please calculate and determine whether overflow occurs. If there is no overflow occurred, writing down the machine code of Z. ___溢出___________

解：X/2 即将 11110100 算数右移一位，得 11111010；2Y 即将 Y 左移一位，得 01100000 发生溢出。

2.7 Execute the following C program, y = ___32769___________

```
Short x = -32767;
Unsigned short y = x;
```

解：16 位补码的真值范围：-32768~32767；而-32768 的 16 位补码位：1000_0000_0000_0000，则-32767 的 16 位补码为：1000_0000_0000_0001；

Y = 1000_0000_0000_0001，其为无符号数，8001H=32769。

2.8 For the following RISC-V assembly code, write the corresponding C statement. Assume that the variables f, g, h, i, and j 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. (3%)

```
sub x30, x28, x29
slli x30, x30, 3
add x3,x30,x10
ld x30, 0(x3)
sd x30, 64(x11)
```

解：B[8] = A[i-j];

```
sub x30, x28, x29 // compute i-j
slli x30, x30, 3 // multiply by 8 to convert the word offset to a byte offset
add x3,x30,x10
ld x30, 0(x3) // load A[i-j]
sd x30, 64(x11) // store in B[8]
```

2.9 Assume the following register contents:
t0 = 0x1234567812345678          t6 = 0x00000000AAAAAAAA
For the register values shown above, what is the value of S1 for the following sequence of instructions? (3%)

```
Slli   s1,t6,0x4
Srai   s2,t0,0x8
xor    s1,s1,t0
```

**解: s1 = 0x123456D2B89EFCD8**

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

2024-5-15.

《Computer Organization & Design》 '2023          Page 5 of 6

| 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 |
|---|---|---|
| Bnez rs,Lable | If (rs!=0) goto Lable | Bne rs,x0,L |
| Sltz rd,rs | Rd=(rs<0)?1:0 | Slt rd,rs,x0 |
| Not rd,rs | rd = ~rs | Xori rd,rs,-1 |
| Jal offset | Jump and link | Jal x1,offset |
| Ret | Return from subroutine | Jalr x0,x1,0 |

5 (10%) Implement the following C code in RISC-V assembly. Hint: Remember that the stack pointer must remain aligned on a multiple of 16:

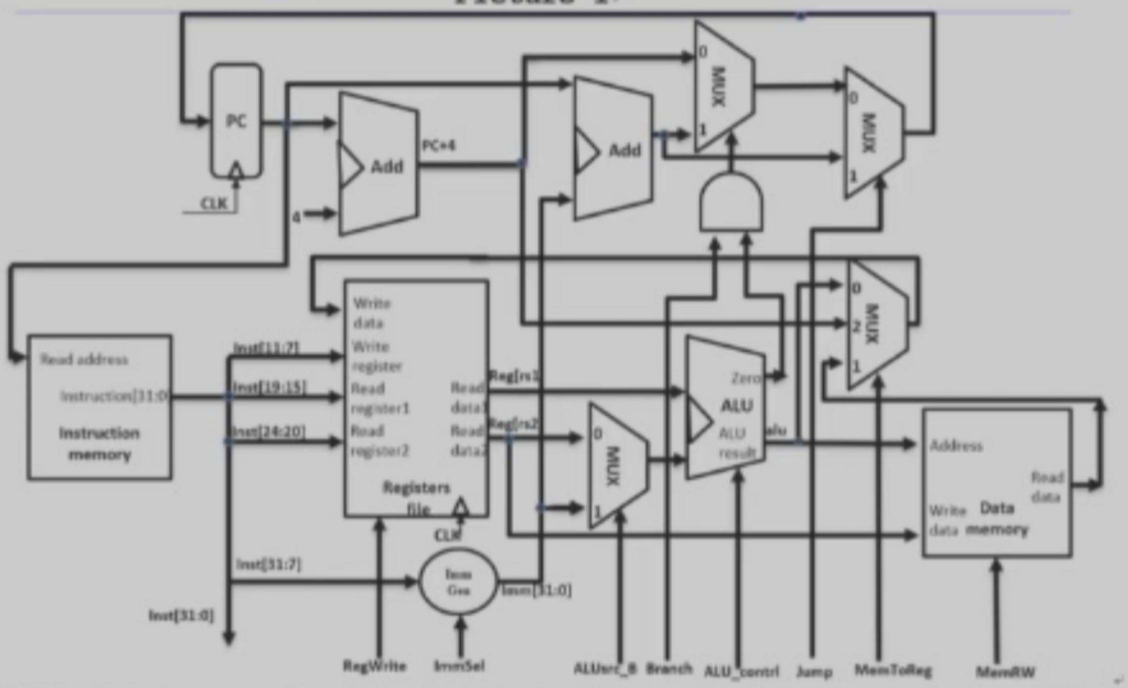
```
int fun(int i){
   if (i==0)
      return 0;
   else if (i == 1)
      return 1;
   else
      return fun(i-1) + fun(i-2);
}
```

参考：

```
fun:
    beq     x10, x0, done      // If i==0, return 0
    addi    x5, x0, 1
    beq     x10, x5, done      // If i==1, return 1
    addi    x2, x2, -16        // Allocate 2 words of stack space
    sd      x1, 0(x2)          // Save the return address
    sd      x10, 8(x2)         // Save the current i
    addi    x10, x10, -1       // x10 = i-1
    jal     x1, fun            // fun(i-1)
    ld      x5, 8(x2)          // Load old i from the stack
    sd      x10, 8(x2)         // Push fun(i-1) onto the stack
    addi    x10, x5, -2        // x10 = i-2
    jal     x1, fun            // Call fun(i-2)
    ld      x5, 8(x2)          // x5 = fun(i-1)
    add     x10, x10, x5       // x10 = fun(i-1)+fun(i-2)
// Clean up:
    ld      x1, 0(x2)          // Load saved return address
    addi    x2, x2, 16         // Pop two words from the stack
done:
    jalr    x0, 0(x1)
```

**6.** (20%) Examine the difficulty of adding a proposed
**ss rs2, rs1, imm** (Store Sum) instruction to RISC-V.
Interpretation: **Mem[Reg[rs2]]=Reg[rs1]+immediate**

*Picture 1*




1) (3%) Which new functional blocks (if any) do we need for this instruction?

some additional muxes

1) (3%) Which new functional blocks (if any) do we need for this instruction?

**some additional muxes**

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

**No functional blocks need to be modified**

3) (8%) Design datapath: Modify the picture1 to demonstrate an implementation of this new instruction.

4) (7%) Design control: to set new signals to support this instruction;

**参考: Dmem_a Dmem_d**