武汉大学计算机学院

2021-2022 学年第一学期 2020 级卓越工程师班/人工智能专业

《计算机组成与设计》

期末考试试题 A 卷（闭卷）

学号_____ 班级 _____ 姓名_____ 成绩_____

注意：所有答题内容必须写在答题纸上，凡写在试题或草稿纸上的一律无效。

本考试使用的 RISC-V 核心指令格式如下：

	31 27	26 25	24 20	19 15	14 12	11 7	6 0
R	funct7		rs2	rs1	funct3	rd	opcode
I	imm[11:0]			rs1	funct3	rd	opcode
S	imm[11:5]		rs2	rs1	funct3	imm[4:0]	opcode
SB	imm[12|10:5]		rs2	rs1	funct3	imm[4:1|11]	opcode
U	imm[31:12]					rd	opcode
UJ	imm[20|10:1|11|19:12]					rd	opcode

一、 性能计算（共 10 分）

若有两种机器采用了不同方法来设计条件分支指令：

（a）P1：通过比较指令设置条件码，然后对条件码进行测试以决定分支。

（b）P2：分支指令中包含了比较操作。

在两种机器中，除了条件分支指令需要 2 个时钟周期之外，其他所有指令都只需 1 个时钟周期。原始的待执行指令中分支指令占 20%，对于 P1，需增加与分支指令数量相同的比较指令；而对于 P2，则不需要。已知 P2 的时钟周期是 P1 的 1.25 倍，那么：

（1）P1 的 CPI 是多少？（4 分）

（2）P2 的 CPI 是多少？（4 分）

（3）哪一个机器更快？（2 分）

解答：

（1）P1 中分支指令需 2 个时钟周期，占比 20%，新增与分支指令对应的比较指令（1 个时钟周期）20%，其他指令需 1 个时钟周期，占比 80%：

CPI1=0.2*2+0.2*1+0.8*1=1.4

（2）P2 中分支指令占比为 20%，其他指令占比 80%：

CPI2=0.2*2+0.8*1=1.2

（3）CPU 时间=IC*CPI*时钟周期，IC1=1.2*IC2，且 T2=1.25*T1，代入表达式：

CPU 时间 1=1.2*IC2*1.4*时钟周期 1=1.68*IC2*T1

CPU 时间 2=IC2*1.2*1.25*T1==1.5*IC2*T1

所以 P2 更快

二、 指令系统（共 27 分）

1、（20 分）有如下 C 语言程序：

```
int foo()
{
   long long D[4*M];
   int i, j;

   for (i=0; i<N; i++) {
      j=0;
      while (j<M) {
         D[4*j]=i+j;
         j+=1;
      }
   }

   return 0;
}
```

下面的代码是其对应的 RV64 汇编程序。

```
foo():
        addi    sp,sp,-672
        sd      x8,664(sp)
        addi    x8,sp,16
.L5:
        li      x5,0
        li      x7,9
        __1__   x5,x7,__2__
.L4:
        li      x6,0
        li      x10,19
        __3__   x6,x10,__4__
        __5__   x11,x5,x6
        slli    x12,x6,__6__
        addi    x12,x8,x12
        __7__   __8__,0(x12)
        addi    x6,x6,1
        jal     x0,__9__
.L3:
        addi    x5,x5,1
        jal     x0,__10__
.L2:
        li      x10,__11__
        ld      x8,664(sp)
        addi    sp,sp,672
        __12__  x0,0(x1)
```

1) 请从上述汇编代码推断出 M 和 N 的值。（2 分）（M=20，N=10）
2) li 是汇编伪指令，是将一个寄存器的值设置为一个常数，在 RV64 中，可以用哪条指令来实现 li x7, 9。（2 分）（addi x7, x0, 9）
3) 填写出汇编代码中缺失的 12 处内容。（12 分）

(1)	bgt	(2)	.L2	(3)	bgt	(4)	.L3
(5)	add	(6)	5	(7)	sd	(8)	x11
(9)	.L4	(10)	.L5	(11)	0	(12)	jalr

4) 汇编代码中第 3 行的作用是什么？为什么需要它？（4 分）（保存寄存器 x8 的值，因为 x8 是被调用者保存寄存器）

2、（7 分）考虑下列 RISC-V 代码（参数传递和结果传递均使用 x10）：

```
    func:
1           addi sp,sp,-16
2           sd   x1,8(sp)
3           sd   x10,0(sp)
4           addi x5,x10,-1
5           bge  x5,x0,L1
6           addi x10,x0,X0
7           addi sp,sp,16
8           jalr x0,0(x1)
9     L1:   addi x10,x10,-1
10          jal  x1,func
11          addi x6,x10,0
12          ld   x10,0(sp)
13          ld   x1,8(sp)
14          addi sp,sp,16
15          mul  x7,x10,x10
16          add  x10,x6,x7
17          jalr x0,0(x1)
```

（1）假设 C 函数的原型为 int func(int a)，请根据汇编代码写出等价的 C 语言代码。（3 分）

（2）第一次执行 addi x6, x10, 0 后，寄存器 x6 的值为多少？ （2 分）

（3）假设传递给 func 函数的实际参数的初始值 4，则循环结束的时候，函数的最后返回结果是多少？（2 分）

参考答案：

（1）long long int func(long long int n)

```
{
    if (n < 1)
        return 0;
    else
        return n*n + func(n-1);

}
```

（2）0

（3）4*4 + 3*3 + 2*2 + 1*1 =30

三、 运算器（共 10 分）

假设有一个虚构的 8 位浮点数标准，称为“minifloat”（如：S **E** EEMMMM，其中符号字段 1 位，指数字段 3 位，尾数字段 4 位），其它属性和 IEEE754 标准一样（如：偏阶，非规格化数值，∞， NaN，等等）。

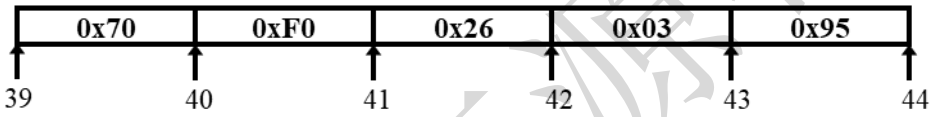
（1）请问偏阶是多少？在[1, 4)范围内有多少个 minifloat？（4 分）

3，32

（2）请写出大于 1 的最小 minifloat 数，用十进制数表示。（3 分）

1+1/16

（3）定义一个 minifloat 类型的数组 A[3]，数组在内存的起始地址是 40，如下图所示，请用十进制写出数组每个元素的真值。（3 分）



A[0] = 0xF0 -∞

A[1] = 0x26 （1+1/4+1/8）*2^{2-3} = 11/16

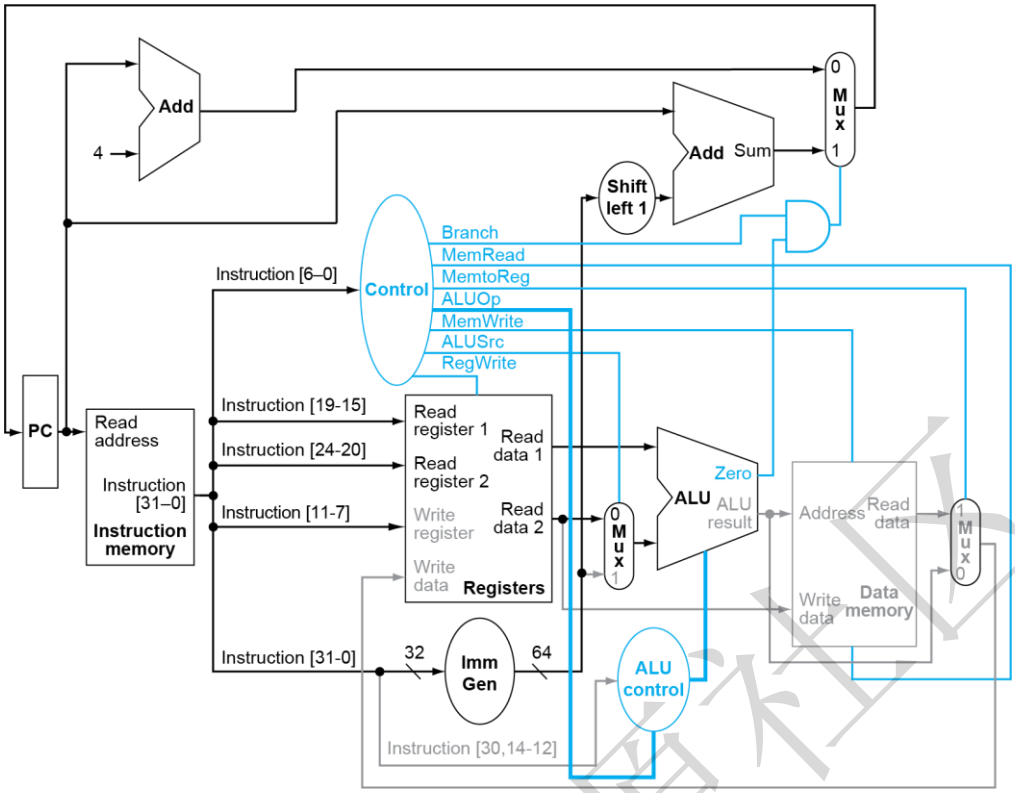
A[2] = 0x03 (0+1/8+1/16)*2^{1-3} = 3/64

四、 CPU（共 22 分）

1、（14 分）为了提高对数据存储器的访问灵活性，现在 RISC-V 中增加一条访存指令，采用“相对基址变址寻址”方式。其指令格式为：mov rd, offset(rs+rt)，这里 rd 为目的寄存器，rs 为基址寄存器，rt 为变址寄存器，offset 为字偏移量，其功能为：

Reg[rd] = Mem[Reg[rs] + Reg[rt] + offset * 4]

现有的单周期 CPU 数据通路如下图：



1) 如果要实现这条 mov 指令，现有的单周期 CPU 数据通路能否满足要求？如果不能，需要新增或修改哪些逻辑模块和控制信号？画出新增的模块和控制信号，明确标注出与现有模块的连接，并解释这些逻辑模块的作用以及控制信号的取值和含义。（10 分）

答：不能满足要求。(1 分)需要增加 1 个“左移 2 位”逻辑模块、1 个“Add”逻辑模块（计算 ALUOut+imm 左移 2 位的和）、1 个“2Mux1”逻辑模块，其输入为 ALUout 和新增 add 模块的输出，输出作为 Memory 地址输入；增加 1 个“MemOffset”控制信号，ld 和 sd 时选择 ALUout，mov 时选择新增 add 模块的输出。

2) 执行新增的 mov 指令时控制单元产生的控制信号分别为什么值？（4 分）

RegWrite	MemRead	ALUSrc	MemWrite	ALU op	MemtoReg	Branch	MemOffset
1	1	0	0	10	1	0	1

2、（12 分）在 RISC-V 五级流水线上执行下列代码：

```
        add rd,rs,rt
        next1
        next2
        beq rs,rt,label
        next3
        next4
        ……
Label:target1
        target2
        ……
```

为处理分支冒险，将分支比较和目标地址计算提前到 ID 级，并且采用预测机制处理分支冒险。如果总是预测分支发生而且配备 BTB（分支目标缓冲区），请在不考虑任何数据冒险的情况下解答下列问题：

（1）如果在 BTB 中没有检索到该指令并且在 ID 阶段确定该指令不是分支指令，该如何？请以 add 指令为例用流水线示意图说明。（3 分）

参考解答：顺序执行

Add rd,rs,rt	IF	ID	EX	MEM	WB		
Next1		IF	ID	EX	MEM	WB	
Next2			IF	ID	EX	MEM	WB

（2）如果在 BTB 中没有检索到该指令并且在 ID 阶段确定该指令是分支指令，又该如何？请以 beq 指令为例用流水线示意图说明。（3 分）

参考解答：

Beq rs,rt,label	IF	ID	EX	MEM	WB		
Next3		IF	ID	EX	MEM	WB	
Next4/Target1			IF	ID	EX	MEM	WB

如果分支实际发生，Next3 应清除掉！

（3）如果在 BTB 中检索到该分支指令并且预测成功，请以 beq 指令为例画出流水线示意图。。（3 分）

参考解答：

Beq rs,rt,label	IF	ID	EX	MEM	WB		
Target1		IF	ID	EX	MEM	WB	
Target2			IF	ID	EX	MEM	WB

（4）如果在 BTB 中检索到该分支指令但是预测失败，该如何？请以 beq 指令为例用流水线示意图说明。（3 分）

参考解答：

Beq rs,rt,label	IF	ID	EX	MEM	WB			
Target1		IF	★	★	★	★		Flushed
Next3			IF	ID	EX	MEM	WB	

五、 内存系统（本题 27 分）

1、(17 分) 假设计算机 M 的主存地址为 24 位，按字节编址；采用分页存储管理方式，虚拟地址为 32 位，页大小为 4KB；TLB 采用 2 路组相联方式和 LRU 替换策略，共 8 组。请回答下列问题。

（1）M 的虚拟地址中哪几位是 TLB 标记？哪几位是 TLB 组号？如果将 M 中的虚拟地址位数增加到 36 位，则 TLB 标记和组号的位数分别是多少位？（4 分）

高 17 位为 TLB 标记；随后 3 位为 TLB 组号；
高 21 位为 TLB 标记；随后 3 位为 TLB 组号；

（2）假设 TLB 初始时为空，访问虚页号依次为 10、12、16、7、20、4、12、16、20、12，在此过程中，哪一个虚页号对应的 TLB 表项被替换？说明理由。（4 分）

虚页号 12 对应的 TLB 表项被替换。因为虚页号 10、12、16、7、20、4、12、16、20 和 12 映射到 TLB 组号依次是 2、4、0、7、4、4、0、0、4、4...，只有映射到 4 号组的虚页号数量大于 2，相应虚页号依次是 12、4 和 20，根据 LRU，当访问第 20 页时，虚页号 4 对应的 TLB 表项被替换出来。

（3）如果将 TLB 改成全相联映射，命中率会有所提高吗？说明理由。如果能提高，但是可能会带来什么负面影响？如果不能，设想一下有什么其他方法能提高命中率。（3 分）

能。工作集只有 6 个表项，如果使用全相联映射，能全部放入 TLB 中，提高命中率。
但是这样可能会提高 TLB 命中时间，可能会提高一点能耗。

（4）在该机器上运行一道程序，采用单级页表，部分页表如下表，请将下列虚拟地址转换成物理地址，写出计算过程，所有数字均为十进制，每项的起始编号是 0。（6 分）

虚拟地址：0793,9048，12862。

有效位	虚拟页号	物理页号	…
1	0	1	…
1	1	3	…
0	2	-	…
1	3	0	…
1	4	2	…
0	5	-	…

0793：虚拟页号：0，页内偏移地址：793　物理地址：4889
9048：虚拟页号：2，页内偏移地址：856　物理地址：-
12862：虚拟页号：3，页内偏移地址：574　物理地址：574

2、（10 分）你正在设计处理器的 cache 系统，假设已经准备采用 write-allocate（写分配）策略，块大小为 64B，所有的 load 和 store 都是针对 8 字节的位置进行的。总线支持 8B 和 64B 的事务处理，也就是从内存既能读写 8B 数据也能读写 64B 的数据。

（1）你不确定应该使用 write-back（写回）还是 write-through（直写）策略中的哪一种。于是你运行了一个 benchmark 程序，它每秒产生 10 亿次写操作而没有读操作，你的 cache 的命中率为 X，那么在程序稳定时，两种写策略的写带宽（bytes/second）分别应该为多少？（4 分）

解：写回：$10^9 * (1-X) * 64$ B/s （2 分）

直写：$10^9 * 8$ B/s （2 分）

（2）如果想使得写带宽的数值（B/s）尽可能的小，X 的值在什么范围时，你会更倾向于使用直写 cache？（2 分）

解：x < 87.5% （2 分）

（3）假设现在你已经决定采用写回策略了，你想研究块大小对 cache 性能的影响。如果你的主要应用总是采用顺序访问的方式访问地址空间上连续的数据，那么你应该选择较大的块还是较小的块？或者说都没有影响？如果是随机访问呢？所有答案都需要给出你的分析和理由。（4 分）

解：连续访问：较大的块，读入一块所需的时间增加，但是能获得更高的空间局部性，读入一个块后续能获得更多的命中，提高命中率。

随机访问：较小的块，读入一块的时间降低，也就是降低了访问一个数据所需要的时间。

随机访问，空间和时间局部性差，无法提高命中率，只能降低（不命中时的）访问时间。