

上海交通大学试卷 (A 卷)

(2021 至 2022 学年 第 2 学期)

班级号 CS2305-1 学号 姓名

课程名称 计算机系统结构 成绩

一、 选择题 (40 分)
请在 Canvas 在线答题

二、 综合题 (60 分) 请在答题纸上答题, 扫描上传 Canvas

1、说明以下这段 RISC-V 代码的功能, 用 C 语言代码来表达其功能。(6 分)

```
addi s0, x0, 0    #x0的值永远为0
addi s1, x0, 1
addi t0, x0, 20
loop:
beq s0, t0, exit
add s1, s1, s1
addi s0, s0, 1
jal x0, loop
exit:
```

2、采用 IEEE 754 单精度浮点数格式计算下题, 给出计算过程。(6 分)

假设有两个实数 x 和 y , $x = -68$, $y = -8.25$, 它们在 C 语言中定义为 float 型变量, x 和 y 分别存放在寄存器 A 和 B 中。另外, 还有一个寄存器 C。A、B、C、都是 32 位的寄存器。请回答下列问题: (要求最终结果用十六进制表示)

(1) 寄存器 A 和 B 中的内容分别是什么?

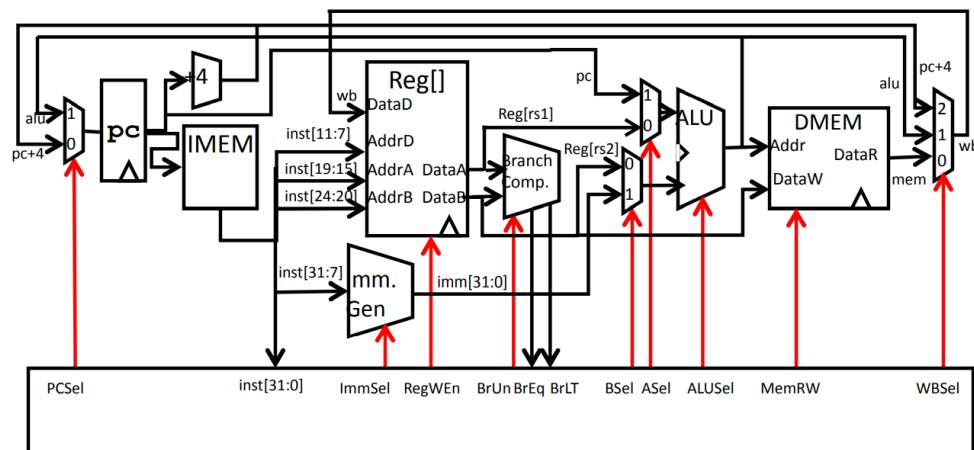
(2) x 和 y 相加后的结果存放在 C 寄存器中, 寄存器 C 中的内容是什么? 请写出计算过程。

3、(6 分) 假定一个分页虚拟存储器系统, 按字节编址, 逻辑地址有 36 位, 页大小为 16KB, 物理地址位数为 32 位, 页表中有效位和修改位各占 1 位、使用位和存取方式位各占 2 位, 而且所有虚拟页都在使用中。请问:

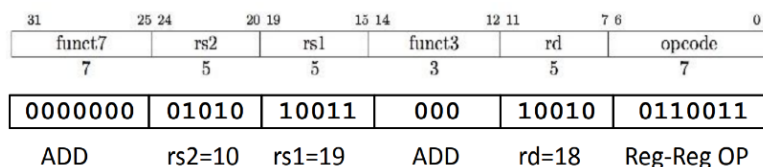
(1) 每个进程的页表大小为多少? 假设页表为一级页表。

(2) 如果所使用的快表 (TLB) 中总的表项数为 256 项, 采用 2 路组相联 Cache 实现, 则 TLB 的大小至少为多少?

表格中已经给出了部分指令应该具备的控制信号。其中 * 表示控制信号可以任意取值；BrUn 为 1 时，表示比较指令的两个比较值是无符号数；ImmSel 取不同的值，是因为不同的指令中的立即数在指令中不同的位置、以及立即数扩展的方法不同。



	BrEq	BrLT	PCSel	ImmSel	BrUn	ASel	BSel	ALUSel	MemRW	RegWEn	WBSel
add	*	*	0 (PC + 4)	*	*	0 (Reg)	0 (Reg)	add	0	1	1 (ALU)
ori	*	*	0	I	*	0 (Reg)	1 (Imm)	or	0	1	1 (ALU)
lw	*	*	0	I	*	0 (Reg)	1 (Imm)	add	0	1	0 (MEM)
sw	*	*	0	S	*	0 (Reg)	1 (Imm)	add	1	0	*
beq	1/0	*	1/0	SB	*	1 (PC)	1 (Imm)	add	0	0	*
jal	*	*	1 (ALU)	UJ	*	1 (PC)	1 (Imm)	add	0	1	2 (PC + 4)
bltu	*	1/0	1/0	SB	1	1 (PC)	1 (Imm)	add	0	0	*



5、(1) 找出下述指令之间的数据相关性 (7 分)

- ① Loop: L.D F0, 0(R1) #读数据存入浮点寄存器 F0
- ② MUL.D F0, F0, F2 #浮点乘法结果写入浮点寄存器 F0
- ③ L.D F4, 0(R2) #读数据存入浮点寄存器 F4
- ④ ADD.D F0, F0, F4 #浮点加法结果写入浮点寄存器 F0
- ⑤ S.D F0, 0(R2) #将浮点寄存器 F0 写入内存单元
- ⑥ SUBI R1, R1, #8 #整数运算, 更新数组下标
- ⑦ SUBI R2, R2, #8 #整数运算, 更新数组下标
- ⑧ BNEZ R1, LOOP #若数组下标不为 0, 跳转回 loop 继续循环语句

(2) (7 分)

接下来, 我们考虑一个单发射、具有前向通路的按序流水线 (fully-bypassed, single-issue, in-order pipeline) 如下图所示, 前向通路未画出。Load 指令读内存的结果要在 X3 流水段的结尾才能获得, 浮点乘法的结果要在 X4 流水段的结尾才能获得。Store 指令要写入内存的数据要在 X2 流水段时准备好即可, 在 X2 流水段前不需要该数据。运算指令例如: MUL 和 SUBI 指令的运算数据必须在 X1 流水段时准备好, 整数运算指令只需要一个周期就能完成。假设分支指令的转移预测总是正确的 (没有因为 branch 指令导致的流水线停顿)。

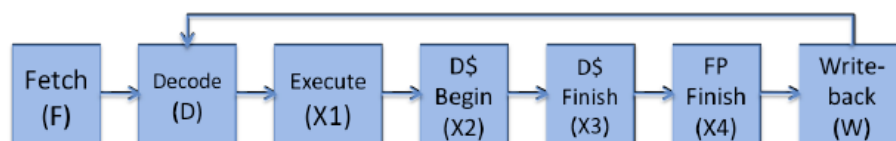


Figure 1. In-order pipeline. Full bypassing is not shown.

写出第一次循环的指令执行时空图。

给出第 (1) 小题中 I1~I8 指令第一次循环在流水线中的执行情况。在指令执行时空图中, 给出各指令所处于的流水段的名称 (用 F、D、X1、X2、X3、X4 和 W 表示各流水段)。

6、一个四路组相联映射的高速缓存（ 4-way associative cache ）容量为 256 bytes ，每个高速缓存块大小为 32 字节（32 bytes per block）。假定：

✧ int 型数据的长度为 4 个字节(4 bytes)，long 型数据长度为 8 个字节(8bytes)。

✧ table 数组的起始地址为 0x0。

✧ 数组在内存中（以行主序）按行存储。

以下两段代码的高速缓存失效率各自为多少？简单给出原因。（8 分）

代码（1）

```
int i;
int j;
int table[4][8];
for (j = 0; j < 8; j++)
    { for (i = 0; i < 4; i++)
        { table[i][j] = i + j; }
    }
```

代码（2）

```
int i;
int j;
long table[4][8];
for (j = 0; j < 8; j++)
    { for (i = 0; i < 4; i++)
        { table[i][j] = i + j; }
    }
```

7、分析以下每一段代码，它是否和对应的串行代码有相同的功能，并且比串行代码快？如果是错误的或者并没有比串行代码加速，请说明原因。（6 分）

（1）

// 将 arr 数组元素设置为斐波那契数列：Set arr to be an array of Fibonacci numbers.

```
arr[0] = 0;
arr[1] = 1;
#pragma omp parallel for
for (int i = 2; i < n; i++)
    arr[i] = arr[i-1] + arr[i - 2];
```

（2）

```
// 将 arr 数组的每一个元素减去其数组下标值
//Decrements element i of arr, n is a multiple of omp_get_num_threads()
#pragma omp parallel {
    int threadCount = omp_get_num_threads();
    int myThread = omp_get_thread_num();
    for (int i = 0; i < n; i++) {
        if (i % threadCount == myThread) arr[i] -= i;
    }
}
```

8、（6 分）

假设程序的核心模块中有两条分支指令，该模块将会被执行成千上万次，在其中一次执行过程中，这两条分支指令的实际执行情况如下（T: Taken; N: not Taken）。

分支指令 1（B1）：T - N - T - N - T - N。

分支指令 2（B2）：T - T - N - T - T - N - T。

假定各个分支指令在每次模块执行过程中的执行情况都一样，并且动态预测时，每个分支指令都有各自的预测表项，每次执行时的初始预测位都相同。

请给出将二位动态预测方案用于预测上述分支指令时，各自的预测正确和错误的次数。

假设二位动态预测，初始预测弱转移（Weak Taken）。状态转换如图所示：

