

计算机组成原理408大题训练营——DAY11

前言概述

大家好，欢迎来到蓝蓝星球组织的第一期计算机组成原理408大题专项突破！

我们将通过计算机组成原理的精选出的一系列重点题以及09-22年所有的真题的练习。针对大家头疼的浮点数计算、cache访存与虚拟存储、一堆和一条指令的运行情况以及I/O数据传输的所有方面帮助大家攻克难关。本次活动深度剖析了历年真题，精选出了需要大家重点掌握的计组细节题目，在听完咸鱼强化的基础上，以真题为始，结合所选的重点题目来全面加强计组大题理解，最后通过二刷真题来全面理解考试出题类型，一共50多道大题帮助大家稳扎稳打，拿下计组大题。本着参加打卡活动希望大家都可以学有所成的初心，邀请了猫叔、酒、Tina等几位同学给大家答疑并且帮助督促大家做好知识的输出工作，希望大家可以认真做题，坚持在星球打卡，念念不忘，必有回响！

题外话

欢迎大家多多关注蓝蓝B站首页：[蓝蓝希望你上岸呀B站首页](#)

关于蓝蓝计算机考研3000+圈子：[计算机考研必备](#)

以及蓝蓝公众号：[应用题训练营专题](#)

也可以关注一下猫叔的B站账号，希望与大家共同进步[薛定谔的猫叔叔是你](#)

做题须知

- 1.建议先听咸鱼强化可后，针对咸鱼所讲真题，先跟着浏览真题，确保了解真题出题难度，浏览题目后，自己去先独立思考题目的知识点是否可以用自己的语言论述并写出
- 2.通过翻阅资料查阅真题的考点，并加以理解，接着利用重点题目的辅助练习来巩固每一章知识点
- 3.针对不会的内容需要反复思考，查阅王道书中相关章节知识，及时巩固题目细节考察重点，归纳总结常考题目类型
- 4.汇总每天的习题成册并留出足够的空白空间方便后期复盘与增补知识点，加强记忆
- 5.持之以恒，多总结多思考，多与管理员和群友及时交流处理所遇到的问题，学习中复盘，复盘中学习，通过培养费曼学习法让自己从输入者变成输出者，手中无剑而心中有剑，万变不离其宗，遇到陌生问题依旧可以迎刃而解的境界！
- 6.以终为始，通过前期的了解真题，到后面的重点模拟，剖析真题，把握出题规律，最后二刷真题，确保题目知识点胸有成竹，闲庭信步，信手捏来！

指令专题相关题目：

01、汇编指令理解，理解条件转移指令和switch的汇编对应内容的计算方法

8. 阅读下面分支结构语句的C语言代码与使用跳转表实现分支结构的汇编代码,回答问题。

C语言代码:	汇编代码及参数说明:
void switcher long a, long b,	a in %rdi, b in %rsi, c in %rdx,
long c, long *dest {	dest in %rcx
long val;	switcher:
switch(a) {	cmpq \$7, %rdi
case 5:	ja .L2
c = b ^ 15;	jmp *.L4(, %rdi, 8)
/* fall through */	.section .rodata
case 0:	.L7: xorq \$15, %rsi
val = c + 112;	movq %rsi, %rdx
break;	.L3: leaq 112(%rdx), %rdi
case 2:	jmp .L6
case 7:	.L5: leaq (%rdi, %rsi), %rdi

val = (c + b) << 2;	salq \$2, %rdi
break;	jmp .L6
case 4:	.L2: movq %rsi, %rdi
val = a;	.L6: movq %rdi, (%rcx)
break;	ret
default:	跳转表:
val = b;	.L4:
}	.quad .L3
*dest = val;	.quad .L2
}	.quad .L5
	.quad .L2
	.quad .L6
	.quad .L7
	.quad .L2
	.quad .L5

- (1) 假设入参 $a=5, b=1, c=1$, 程序执行后内存 `dest` 处值为多少? 假设入参 $a=9, b=1, c=1$, 程序中哪些分支被执行? 这种情况下 `cmp` 指令执行后, `CF`、`ZF` 的内容是什么? 为什么编译器在第 2 行汇编代码将 `a` 与 7 进行比较?
- (2) 上述代码中, 哪条指令是条件转移指令? 哪几条指令一定会使程序跳转执行? 这些指令对指令流水线有什么影响?
- (3) 第 8 行指令 `xorq` 实现了 `b` 和 15 的异或运算, 如果 `b` 是 `double` 类型能否使用 `xorq` 指令实现这个操作? 为什么?
- (4) 简要描述上述代码如何使用跳转表实现 `switch-case` 语句? 这样的实现有什么好处?

02、根据MIPS汇编指令以及五级流水线来判断指令时间是否发生数据相关，以及在不采用和采用转发技术的情况下加入几条指令才可以避免冒险

```
I1: add    $t1, $s1, $s0
I2: sub    $t2, $s0, $s3
I3: add    $t1, $t1, $t2
```

假定某处理器使用“取指 (IF)、译码/取数 (ID)、执行 (EX)、访存 (MEM)、写回 (WB)”的 5 段流水线，若在此处理器上运行上面的汇编指令序列，请回答下列问题：

- (1) 这段指令序列中，哪些指令之间发生数据相关？
- (2) 如果不使用“转发”技术，需要在这段指令序列中哪里加入空操作指令 (nop) 才能避免发生数据冒险，需要加入几条空操作指令？
- (3) 若使用“转发”技术且不插入空操作指令，是否可以解决数据冒险？如果不能，需要在哪里、加入多少条空操作指令才能避免数据冒险？

03、根据MIPS汇编指令以及五级流水线来判断指令时间是否发生数据相关，以及在不采用和采用转发技术的情况下加入几条指令才可以避免冒险以及其CPI是多少

20. 下面是一段 MIPS 指令序列：

```
I1: add    $s2, $s1, $s0
I2: sub    $t0, $s0, $s2
I3: lw     $t1, 4($t0)
I4: add    $t1, $t1, $t0
```

假定某处理器使用“取指、译码/取数、执行、访存、写回”的5级流水线，并在此处理器上运行该汇编指令序列。在此流水线数据通路中，使用双端口寄存器，且寄存器的写操作和读操作分别安排在一个时钟周期内的前半周期和后半周期并独立工作。请回答下列问题：

- (1) 此段指令序列中，哪些指令之间存在数据相关？
- (2) 如果不采用“转发”技术，为了避免在这段指令序列的执行过程中发生数据冒险，需要在哪里、加入多少条 nop 指令？nop 指令增加的百分比为多少？执行完成这段指令序列总共需要多少个时钟周期？
- (3) 若仅使用“转发”技术，是否完全可以解决数据冒险？如果不行，需要在哪里、加入多少条 nop 指令才能避免发生数据冒险？
- (4) 如果通过硬件阻塞解决数据冒险，那么在不使用“转发”和使用“转发”两种情况下，执行序列中的4条指令的CPI分别是多少？

需在第4条指令前插入1条nop指令

共需 $5 + (4 + 1 - 1) = 9$ 个时钟周期

不用转发需 14 个时钟周期

$$CPI = 14/4 = 3.5$$

用转发需 9 个时钟周期

$$CPI = 9/4 = 2.25$$

(1) 第1、2条存在 $s2$ 的数据相关 第2、3指令存在 $t0$ 的数据相关
第2、4条存在 $s0$ 的数据相关 第3、4条存在 $t1$ 的数据相关

(2) 不用转发，需在第2、3、4条指令前均插入2条nop指令

共插入6条nop指令来解决数据冒险，总条数6条nop增加150%

指令序列执行时钟周期为 $5 + (4 + 6 - 1) = 10$

(3) 使用转发技术 第1、2与2、3与2、4之间数据相关可解决

但第3、4条指令之间为 Load-Use型，转发无法解决

04、读题目C语言和汇编代码与指针的偏移变化，理解数组在机器语言大小的表示以及函数执行过程是否缺页，学会提炼共同点，总结题目设问

24. 一段C语言程序代码如下所示。

```
#include<stdio.h>
#define N 4
int s=0;
int arr[N]={-591, -126, -1, 60};
int get_sum() {
    int i;
    for(i=0;i<N;++i)
        s=s+arr[i];
    return s;
}
void main(){
    printf("The sum of arr is %d\n",get_sum());
}
```

在某32位计算机上，数据采用小端对齐方式存储，用GCC编译上述源程序，生成的可执行文件名为test，然后反汇编得到get_sum函数的结果如下（提示：此汇编指令中\$表示常量，加%表示寄存器）。

08048440: 55	push %ebp
08048441: 89 e5	mov %esp,%ebp
08048443: 83 ec 10	sub \$0x10,%esp
08048446: c7 45 fc 00 00 00 00	movl \$0x0,-0x4(%ebp)
0804845d: eb 1a	jmp 08048469<sum+0x29>
0804845f: 8b 45 fc	mov -0x4(%ebp),%eax
08048452: 8b 14 05 ab 97 04 08	mov 0x80497ab(,%eax,4),%edx
08048459: a1 af 97 04 08	mov 0x80497af,%eax
0804845e: 01d0	add %edx,%eax
08048460: a3 af 97 04 08	mov %edx,0x80497af
08048465: 83 45 fc 01	addl \$0x1,-0x4(%ebp)
08048469: 83 7d fc 03	cmpl \$0x3,-0x4(%ebp)
0804846d: 7e e0	jle 0804845f<sum+0xf>
0804846f: a1 f0 96 04 08	mov 0x80497af,%eax
08048474: c9	leave
08048475: c8	ret

请回答下列问题：

- (1) 数组arr的首地址为0x080497ab，那么0x080497ab、0x080497ad这两个存储单元存储的内容分别是多少（用十六进制表示）？
- (2) get_sum函数对应的机器代码有多少字节？0804845d处的指令是什么类型的指令？此指令对于指令流水线有何影响？
- (3) 如果页表大小为4KB，那么get_sum函数对应的机器代码占据几页？页号分别为多少？
- (4) 在执行get_sum函数的过程中，访问指令和访问数据时各出现了几次缺页？

05、读题目汇编代码在五级流水线的情况下，理解指令在执行过程中如果出现数据相关如果处理数据相关以及总共需要的时钟周期是多少。类比12年真题

25. 某计算机使用“取指令（IF）、译码/读寄存器（ID）、执行/有效地址计算（EX）、存储器访问（MEM）、结果写回寄存器（WB）”五段流水线。现在有下列指令序列进入该流水线：

```
I1: SUB    R1, R2, R3    ; (R2)-(R3)→R1
I2: ADD    R4, R1, R5    ; (R1)+(R5)→R4
I3: OR     R6, R1, R7    ; (R1) AND (R7)→R6
I4: AND    R8, R1, R9    ; (R1) OR (R9)→R8
I5: XOR    R10, R1, R11 ; (R1) XOR (R11)→R10
```

请回答下列问题：

- (1) 如果不处理指令之间的数据相关，允许这些指令直接进入流水线，请问哪几条指令会从寄存器 R1 中读取到不正确的操作数？
- (2) 如果使用“将指令 2 推迟到所需操作数被写回到寄存器后再执行”的方式来避免数据相关的问题，那么处理器需使用多少个时钟周期才能执行完此指令序列？