

北京大学信息科学技术学院考试试卷

考试科目： 计算机系统导论 姓名： _____ 学号： _____

考试时间： 2013 年 11 月 12 日 任课教师： _____

题号	一	二	三	四	五	六	七	八	总分
分数									
阅卷人									

北京大学考场纪律

1、考生进入考场后，按照监考老师安排隔位就座，将学生证放在桌面上。无学生证者不能参加考试；迟到超过 15 分钟不得入场。在考试开始 30 分钟后方可交卷出场。

2、除必要的文具和主考教师允许的工具书、参考书、计算器以外，其它所有物品（包括空白纸张、手机、或有存储、编程、查询功能的电子用品等）不得带入座位，已经带入考场的必须放在监考人员指定的位置。

3、考试使用的试题、答卷、草稿纸由监考人员统一发放，考试结束时收回，一律不准带出考场。若有试题印制问题请向监考教师提出，不得向其他考生询问。提前答完试卷，应举手示意请监考人员收卷后方可离开；交卷后不得在考场内逗留或在附近高声交谈。未交卷擅自离开考场，不得重新进入考场答卷。考试结束时间到，考生立即停止答卷，在座位上等待监考人员收卷清点后，方可离场。

4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准交头接耳，不准偷看、夹带、抄袭或者有意让他人抄袭答题内容，不准接传答案或者试卷等。凡有违纪作弊者，一经发现，当场取消其考试资格，并根据《北京大学本科考试工作与学术规范条例》及相关规定严肃处理。

5、考生须确认自己填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。

学校倡议所有考生以北京大学学生的荣誉与诚信答卷，共同维护北京大学的学术声誉。

以下为试题和答题纸，共 14 页。

得分

第一题 选择题（每小题 2 分，共 34 分）

（注：每小题有一个或多个正确答案）

1、变量 x 的值为 0x01234567，地址 $\&x$ 为 0x100；则该变量的值在 x86 和 Sun 机器内存中的存储排列顺序正确的是（ ）

选项	机器类型	地址			
		0x100	0x101	0x102	0x103
A	x86	67	45	23	01
	Sun	01	23	45	67
B	x86	76	54	32	10
	Sun	01	23	45	67
C	x86	01	23	45	67
	Sun	67	45	23	01
D	x86	01	23	45	67
	Sun	01	23	45	67

2、假设下列 int 和 unsigned 数均为 32 位，

$\text{int } x = 0x80000000;$

$\text{unsigned } y = 0x00000001;$

$\text{int } z = 0x80000001;$

以下表达式正确的是（ ）

A. $(-x) < 0$

B. $(-1) > y$

C. $(z < 3) == (z * 8)$

D. $y * 24 == z < 5 - z < 3$

3、对 $x = 1\frac{1}{8}$ 和 $y = 1\frac{3}{8}$ 进行小数点后两位取整（rounding to nearest even），结果正确的是（ ）

A. $1\frac{1}{4}, 1\frac{1}{4}$ B. $1, 1\frac{1}{4}$ C. $1\frac{1}{4}, 1\frac{1}{2}$ D. $1, 1\frac{1}{2}$

4、在完成本课程的 Bomb Lab 的时候，通常先执行 gdb bomb 启动调试，然后执行 ____ explode_bomb 命令以防引爆炸弹，之后在进行其他必要的设置后，最后执行 ____ 命令以便开始执行程序。上述两个空格对应的命令是（ ）

A. st, ru B. br, go C. br, ru D. st, go

5、已知函数 `int x(int n) { return n*____; }` 对应的汇编代码如下：

```
lea(%rdi,%rdi,4),%rdi
lea(%rdi,%rdi,1),%eax
retq
```

请问横线上的数字应该是 ()

- A. 4 B. 5 C. 2 D. 10

6、32 位 x86 计算机、Windows 操作系统下定义的一个 `structure S` 包含三个部分：`double a, int b, char c`，请问 `S` 在内存空间中最多和最少分别能占据多少个字节（32 位 Windows 系统按 1、4、8 的原则对齐 `char`、`int`、`double`）？答：()

- A. 16, 13
B. 16, 16
C. 24, 13
D. 24, 16

7、x86 体系结构的内存寻址方式有多种格式，请问下列哪些指令是正确的：()

- A. `movl $34, (%eax)`
B. `movl (%eax), %eax`
C. `movl $23, 10(%edx,%eax)`
D. `movl (%eax), 8(%ebx)`

8、x86 体系结构中，下面哪些选项是错误的？答：()

- A. `leal` 指令只能够用来计算内存地址
B. `x86_64` 机器可以使用栈来给函数传递参数
C. 在一个函数内，改变任一寄存器的值之前必须先将其原始数据保存在栈内
D. 判断两个寄存器中值大小关系，只需 `SF`（符号）和 `ZF`（零）两个条件码

9、下面对 `RISC` 和 `CISC` 的描述中，错误的是：()

- A. `CISC` 指令系统中的指令数目较多，有些指令的执行周期很长；而 `RISC` 指令系统中通常指令数目较少，指令的执行周期都较短。
B. `CISC` 指令系统中的指令编码长度不固定；`RISC` 指令系统中的指令编码长度固定，这样使得 `RISC` 机器可以获得了更短的代码长度。
C. `CISC` 指令系统支持多种寻址方式，`RISC` 指令系统支持的寻址方式较少。
D. `CISC` 机器的寄存器数目较少，函数参数必须通过栈进行传递；`RISC` 机器的寄存器数目较多，可以通过寄存器来传递参数，避免了不必要的存储访问。

- 10、下面对流水线技术的描述，正确的是：（ ）
- A. 流水线技术不仅能够提高指令的吞吐率，还能减少单条指令的执行时间。
 - B. 不断加深流水线级数，总能获得性能上的提升。
 - C. 流水级划分应尽量均衡，吞吐率会受到最慢的流水级影响。
 - D. 指令间数据相关可能引发数据冒险，可通过数据转发或暂停流水线来解决。

(11-13)、在教材所描述的流水线处理器（the PIPE processor）上分别运行如下四段Y86程序代码。请分析其中数据冒险的具体情况，并回答后续3个小题。

<pre>#Program 1: mrmovl 8(%ebx), %edx rmmovl %edx, 16(%ecx)</pre>	<pre>#Program 2: mrmovl 8(%ebx), %edx nop rmmovl %edx, 16(%ecx)</pre>
<pre>#Program 3: mrmovl 8(%ebx), %edx nop nop rmmovl %edx, 16(%ecx)</pre>	<pre>#Program 4: mrmovl 8(%ebx), %edx nop nop nop rmmovl %edx, 16(%ecx)</pre>

- 11、对于每段程序，请指出是否会因为数据冒险导致流水线停顿（Stall）。
- Program 1: (), Program 2: (), Program 3: (), Program 4: ();
- A. Stall B. No-Stall
- 12、对于每段程序，请指出流水线处理器内是否会产生数据转发（Forwarding）。
- Program 1: (), Program 2: (), Program 3: (), Program 4: ();
- A. Forwarding B. No-Forwarding
- 13、对于每段程序，请指出流水线处理器内使用哪个信号进行数据转发，如果不进行数据转发，则用none表示。
- Program 1: (), Program 2: (), Program 3: (), Program 4: ();
- A. m_valM B. W_valM C. none
- 14、下面哪些选项是错误的？答：（ ）
- A. 同一个任务采用时间复杂度为 $O(\log N)$ 算法一定比采用复杂度为 $O(N)$ 算法的执行时间短
 - B. 编译器进行程序优化时，总是可以使用算数结合律来减少计算量
 - C. 增大循环展开（loop unrolling）的级数，有可能降低程序的执行性能（即增加执行时间）
 - D. 分支预测时，“总是预测不跳转”（branch not taken）一定比“总是预测跳转”（branch taken）预测准确率高

15、以下哪些程序优化编译器总是可以自动进行？（假设 `int i, int j, int A[N], int B[N], int m` 都是局部变量，`N` 是一个整数型常量，`int foo(int)` 是一个函数）答：（ ）

	优化前	优化后
A.	<pre>for (j = 0 ; j < N ; j++) m += i*N*j;</pre>	<pre>int temp = i*N; for (j= 0 ; j < N ; j++) m += temp * j;</pre>
B.	<pre>for (j = 0 ; j < N ; j++) B[i] *= A[j];</pre>	<pre>int temp = B[i]; for (j= 0 ; j < N ; j++) temp *= A[j]; B[i] = temp;</pre>
C.	<pre>for (j = 0 ; j < N ; j++) m = (m + A[j]) + B[j];</pre>	<pre>for (j = 0 ; j < N ; j++) m = m + (A[j] + B[j]);</pre>
D.	<pre>for (j = 0 ; j < foo(N) ; j++) m ++;</pre>	<pre>int temp = foo(N); for (j= 0 ; j < temp ; j++) m ++;</pre>

16、如果直接映射高速缓存大小是 4KB，并且块（block）大小为 32 字节，请问它每组（set）有多少行（line）？答：（ ）

- A. 128 B. 64 C. 32 D. 1

17、关于局部性（locality）的描述，不正确的是：（ ）

- A. 数组通常具有很好的时间局部性
- B. 数组通常具有很好的空间局部性
- C. 循环通常具有很好的时间局部性
- D. 循环通常具有很好的空间局部性

得分

第二题 (8 分)

1) 判断下表中每一行表达式对或错。如果错, 请举出反例或说明原因 (每行 1 分)

int x, y ;
unsigned u, v ;

	True or False	原因或举出反例
if $x < 0$, then $x * 2 < 0$		
$u \leq -1$		
if $x > y$, then $-x < -y$		
if $u > v$, then $-u > -v$		

2) 请按单精度浮点数表示下表中的数值, 首先写出形如 $(-1)^s \times M \times 2^E$ 的表达式, 然后给出十六进制的表示。(每格 1 分)

注: 符号位 (s): 1-bit; 阶码字段 (exp): 8-bit; 小数字段 (frac): 23-bit; 偏置值 (bias): 127。

Value	$(-1)^s \times M \times 2^E, 1 \leq M < 2$	Hex representation
$-1\frac{1}{2}$		
2^{-149}		

得分

第三题 （11 分）

阅读下面的 C 代码：

```
/*
 * Copyright (C) 2013 Davidlohr Bueso <davidlohr.bueso@hp.com>
 *
 * Based on the shift-and-subtract algorithm for computing integer
 * square root from Guy L. Steele.
 */

/**
 * int_sqrt - rough approximation to sqrt
 * @x: integer of which to calculate the sqrt
 *
 * A very rough approximation to the sqrt() function.
 */
unsigned long int_sqrt(unsigned long x)
{
    unsigned long b, m, y = 0;

    if (x <= 1)
        return x;

    m = 1UL << (BITS_PER_LONG - 2);
    while (m != 0) {
        b = y + m;
        y >>= 1;

        if (x >= b) {
            x -= b;
            y += m;
        }
        m >>= 2;
    }

    return y;
}
```

1) 在 64 位的机器上 BITS_PER_LONG 的定义为 long 类型的二进制位数，它是多少位？（1 分）

答：

2) 填写下面反汇编中的缺失的内容：（每空 1 分）

<int_sqrt>:

```

4004c4:    push    %rbp
4004c5:    mov     %rsp,%rbp
4004c8:    mov     %rdi,-0x28(%rbp)

4004cc:    movq    __ (1) _____, -0x8(%rbp)
4004d4:    cmpq    $0x1,-0x28(%rbp)

4004d9:    ja      __ (2) _____ <int_sqrt+??>
4004db:    mov     -0x28(%rbp),%rax

4004df:    jmp     __ (3) _____ <int_sqrt+??>
4004e1:    movl    $0x0,-0x10(%rbp)

4004e8:    movl    __ (4) _____, -0xc(%rbp)

4004ef:    jmp     __ (5) _____ <int_sqrt+??>
4004f1:    mov     -0x10(%rbp),%rax
4004f5:    mov     -0x8(%rbp),%rdx

4004f9:    lea     __ (6) _____,%rax
4004fd:    mov     %rax,-0x18(%rbp)
400501:    shrq    -0x8(%rbp)
400505:    mov     -0x28(%rbp),%rax
400509:    cmp     -0x18(%rbp),%rax

40050d:    jb      __ (7) _____ <int_sqrt+??>
40050f:    mov     -0x18(%rbp),%rax
400513:    sub     %rax,-0x28(%rbp)
400517:    mov     -0x10(%rbp),%rax
40051b:    add     %rax,-0x8(%rbp)

40051f:    shrq    __ (8) _____, -0x10(%rbp)
400524:    cmpq    $0x0,-0x10(%rbp)

400529:    jne     __ (9) _____ <int_sqrt+??>

40052b:    mov     -0x8(%rbp),__ (10) _____
40052f:    leaveq
400530:    retq

```


得分

第四题（10 分）

阅读下面的汇编代码：

<f>:

```

4004c4:    push    %rbp
4004c5:    mov     %rsp,%rbp
4004c8:    sub     $0x10,%rsp
4004cc:    mov     %edi,-0x4(%rbp)
4004cf:    cmpl    $0x1,-0x4(%rbp)
4004d3:    ja      4004dc <f+0x18>
4004d5:    mov     $0x1,%eax
4004da:    jmp     40052d <f+0x69>
4004dc:    mov     -0x4(%rbp),%eax
4004df:    and     $0x1,%eax
4004e2:    test    %eax,%eax
4004e4:    jne     4004f5 <f+0x31>
4004e6:    mov     0x200440(%rip),%eax    # 60092c <x.1604>
4004ec:    add     $0x1,%eax
4004ef:    mov     %eax,0x200437(%rip)    # 60092c <x.1604>
4004f5:    mov     -0x4(%rbp),%eax
4004f8:    and     $0x1,%eax
4004fb:    test    %al,%al
4004fd:    je      40050e <f+0x4a>
4004ff:    mov     0x20042b(%rip),%eax    # 600930 <y.1605>
400505:    add     $0x1,%eax
400508:    mov     %eax,0x200422(%rip)    # 600930 <y.1605>
40050e:    mov     -0x4(%rbp),%eax
400511:    sub     $0x1,%eax
400514:    mov     %eax,%edi
400516:    callq   4004c4 <f>
40051b:    mov     0x20040f(%rip),%edx    # 600930 <y.1605>
400521:    lea     (%rax,%rdx,1),%edx
400524:    mov     0x200402(%rip),%eax    # 60092c <x.1604>
40052a:    lea     (%rdx,%rax,1),%eax
40052d:    leaveq
40052e:    retq

```

1) 程序

```
main()
{
    unsigned int n;
    for (n=1; n<4; n++) {
        printf("f(%d) = %x\n", n, f(n));
    }
}
```

的运行结果为: f(1)=1, f(2)=4e, f(3)=9f, 请填写 f 函数所需要的内容 (每空 1 分):

```
#define N   (1) _____
#define M   (2) _____
```

```
struct P1 {char c[N]; char *d[N]; char e[N]; } P1;
struct P2 {int i[M]; char j[M]; short k[M]; } P2;
```

```
unsigned int f(unsigned int n)
{
    (3) _____ unsigned int x = sizeof(P1);
    (4) _____ unsigned int y = sizeof(P2);

    if ( (5) _____ )
        return 1;

    if ( (6) _____ )
        x++;

    if ( (7) _____ )
        y++;

    return (8) _____;
}
```

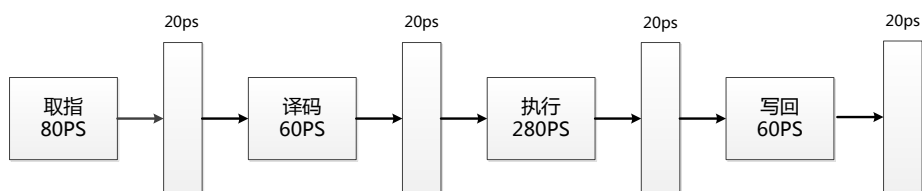
2、程序

```
main()
{
    printf("%x, %x\n", f(2), f(2));
}
```

的运行结果为: _____ (2 分)

得分

第五题（9 分）



在“取指-译码-执行-写回”的四级流水线中，各流水级的工作内容和延迟如上图所示，寄存器的延迟也已标出。数据和指令分别存放在不同的存储器中。Cycle N 写入寄存器文件的数据 Cycle N+1 才可读出。请问：

1) 若不考虑流水线填充和清空时间，请计算该处理器的吞吐率。（1 分）

答：

2) 若将该处理器改造为单周期（SEQ），请计算 SEQ 处理器的吞吐率。（1 分）

答：

3) 在上述流水线中，执行阶段包含了访问数据存储的时间。对于如下 Y86 程序段，指令间存在哪些数据相关（dependence），会引起哪些数据冒险（hazard）？（5 分）

Prog:

```

irmovl $128, %edx      #instr1
irmovl $3, %ecx        #instr2
rmmovl %ecx, 0(%edx)   #instr3
irmovl $10, %ebx       #instr4
mrmovl 0(%edx), %eax   #instr5
addl   %ebx, %eax      #instr6
  
```

答：

4) 以上的数据冒险，可以通过转发（forward）的方法解决。请结合上述程序代码和流水线结构图逐个说明解决方案。（2 分）

答：

得分

第六题（9 分）

请分析Y86 ISA中定义的两条指令（cmovXX、call）和一条新加入Y86 ISA的IA32指令（decl：将操作数减1）。若在教材所描述的SEQ处理器上执行这些指令，请按下表填写每个阶段进行的操作。如果在某一阶段没有任何操作，请填写none指明。（每条指令3分）

注1、所用到的指令编码为：

cmovXX rA, rB	2	fn	rA	rB	
call Dest	8	0	Dest		
decl rA	C	0	rA	F	

注2、需说明的信号包括：icode, ifun, rA, rB, valA, valB, valC, valE, valP; the register file R[], data memory M[], Program counter PC, condition codes CC。

Stage	cmovXX rA, rB	call Dest	decl rA
Fetch			
Decode			
Execute			
Memory			
Write back			
PC update			

得分

第七题（10 分）

已知如下的汇编程序实现了函数 transform(char* src, char* tgt, char delta)

transform:

```

    jmp    L2
L1:
    add    %edx, %eax
    add    $1, %rdi
    mov    %al, (%rsi)
    add    $1, %rsi
L2:
    movzbl (%rdi), %eax
    test   %al, %al
    jne    L1
    movb   $0, (%rsi)
    retq

```

注：x86-64 位指令中传递前三个参数分别使用寄存器%rdi, %rsi 和%rdx

1) 写出 transform 函数对应的 C 语言版本（2 分）

2) 假设读写访存指令延迟为 20 个时钟周期，其他指令延迟为 2 个时钟周期，所有分支预测都成功。同时 CPU 包含足够多的部件来实现指令并行，那么 CPE 最低应该是多少（2 分）？为什么（2 分）？

3) 已知 src 对应字符串中每个字符 c 都满足 $0 < c \leq 80$ 且 $0 \leq \text{delta} \leq 5$ 。通过下面的改写，可以把 transform 程序 CPE 的理论下限降低一半，请填空。假设程序运行在小端法机器上。（每空 1 分）

```

void transform(char* src, char* tgt, char delta) {
    short x = _____;
    while(*src && _____) {
        *(short*)tgt = *(short*)src + x;
        src += 2;
        tgt += 2;
    }
    *(short*)tgt = _____ ? *(short*)src + delta : *(short*)tgt & _____;
}

```

得分

第八题（9 分）

假设存在一个能够存储四个数据的 Cache，每一行（line）的数据块长度（B）为 2 字节。假设内存空间一共是 32 个字节，即内存空间地址长度一共是 5 个比特：从 0（00000₂）到 31（11111₂）。某程序片段一共有 8 个数据读取操作，每个操作的地址按顺序如下所示（单位是字节），

数据访问地址序列：1→4→17→2→8→16→9→0

数据替换采用 LRU（least recently used）策略，即“最久没有被访问”的 cache line 作为替换对象。例如：如果 A、B、C、D 四个 cache line 被访问的顺序是 A→B→A→C→D，则在这四个 cache line 里面，B 是最久没有被访问的 cache line。

1) 如果 Cache 的结构是直接映射（directed mapped）（S=4，E=1），如下图所示。请在下图空白处填入，访问上述数据序列访问后 Cache 的状态。（注：TAG 使用二进制格式，V=1 代表数据有效，用[A-B]表示地址 A 到 B 之间对应的数据）（4 分）

V	TAG	DATA

2) 如果 cache 的结构如下图所示（S=2，E=2），请填入访问后的状态（2 分）

V	TAG	DATA	V	TAG	DATA

在这种情况下，数据访问一共产生了多少次 Miss：_____（1 分）

3) 如果 cache 的结构变成（S=1，E=4），最终存储在 Cache 里面的数据有哪些？（注：只需要填写数据部分，顺序不限）

_____, _____, _____, _____（2 分）