2015 年

一、ADBDA CDCBC CBCCD AADCA

二、1. 1)D　　2)D　　3)B,D　　4) E　　5)A,D

6)E　当两个数绝对值都很小时，相乘以后由于精度不够会变成 0

7)C　　8)C

2. 1) A.128　　B.64

2)

数字	二进制表示
最小的正规格化数	0 0001 0000000
最大的非规格化数	0 0000 1111111
$17\ \frac{1}{16}$	0 1011 0001000
$-\ \frac{1}{8192}$	1 0000 0000001
$10\ \frac{3}{8}$	0 1010 0100110
-∞	1 1111 0000000

三、1. 8

2. up->next->x=*(up->next->p)-(up->y)

3.　(3个错，每个错2分)

X86代码

```
xorl %ecx,%ecx
movl 8(%edx),%ebp        ||应为：  movl 8(%ebp),%edx
movl 12(%ebp),%eax

LOOP:
movl (%eax),%eax
add $1,%ecx
test %ecx,%edx
jne LOOP

movl (%eax),%eax
```

Y86代码

```
mrmovl 8(%edx),%ebp     ||应为：  mrmovl 8(%ebp),%edx
mrmovl 12(%ebp),%eax
irmovl $1,%ecx

LOOP:
mrmovl (%eax),%eax
test %ecx,%edx              ||应为：subl %ecx,%edx
jne LOOP

mrmovl (%eax),%eax
```

4.

A=3 B=7

四、1.M_4[PC+2]

PC+6

valA – valB

（注：也可以是valB - valA）

PC ← valE==0 ? valC : valP

2.2，1

3. D_icode = INewJE & !d_equal

F: normal

D: bubble

E normal

4. F: stall

D: stall

E: bubble

5. stall stall stall stall buuble

五、1. 1） 4

2） 16

3） A, 14

4)

V	Tag	Data	V	Tag	Data
1	101	Array[1][0]~ Array[1][3]	1	100	Array[0][0]~ Array[0][3]
1	110	Array[2][0]~ Array[2][3]	1	111	Array[3][0]~ Array[3][3]

2.

1) 6

2) 4

3) 4

4) C

5)

V	Tag	Data
1	100000	Array[0][0]~ Array[0][3]
V	Tag	Data
1	100101	Array[1][0]~ Array[1][3]
V	Tag	Data
1	101001	Array[2][0]~ Array[2][3]
V	Tag	Data
1	101101	Array[3][0]~ Array[3][3]

2014 年

一、CDBCB BADCD CADBD C

二、

	Decimal Representation	Binary Representation
Z	25	1 1001
y – *z*	0	0 0000
TMin	-16	1 0000

Value	$(-1)^s \times M \times 2^E$ 1<=M<2	Hex representation
0.375	1.1 x 2^{-2}	0x3EC00000
-12.5	(-1) * 1.1001*2^3	0xC1480000

三、

a			
13	52	01	00

b
FB

c
5A

d			
00	40	40	44

四、

```
int f(int n, int m) {
    if (m > 0) {
        if (n > 1) {
            int r = f(n - 1, m);
            return (r - 1 + m) % n + 1;
        }
        else if (n == 1) {
            return 1;
        }
    }
    return 0;
}

mov     %rbp,-0x8(%rsp)
mov     (%rsp),%rbx
```

0x7fffffffe38c	X
0x7fffffffe388	X
0x7fffffffe384	X
0x7fffffffe380	X
0x7fffffffe37c	X
0x7fffffffe378	X
0x7fffffffe374	0x0
0x7fffffffe370	0x00400505
0x7fffffffe36c	0x0
0x7fffffffe368	0x4
0x7fffffffe364	0x0
0x7fffffffe360	0x3
0x7fffffffe35c	0x0
0x7fffffffe358	0x00400505
0x7fffffffe354	0x0
0x7fffffffe350	0x3
0x7fffffffe34c	0x0
0x7fffffffe348	0x3
0x7fffffffe344	0x0
0x7fffffffe340	0x00400505
0x7fffffffe33c	0x0
0x7fffffffe338	0x2
0x7fffffffe334	0x0
0x7fffffffe330	0x3
0x7fffffffe32c	X
0x7fffffffe328	X
0x7fffffffe324	X
0x7fffffffe320	X

五、int fun(unsigned x) {

```
    int bit_sum = 0;
    while (（int）x > 0 ) {
        bit_sum += x % 10 ;
        x = x / 10 ;
    }
    if ( bit_sum % 3 == 0 )
        return 1;
    else
```

```
    return 0;
}
```

六、

Stage	caddXX rA, rB
Fetch	icode:ifun ← M_1[PC] rA:rB ← M_1[PC+1] valP ← PC+2
Decode	valA ← R[rA] valB ← R[rB]
Execute	valE ← valA+valB Cnd ← Cond(CC,ifun)
Memory	none
Write back	if(Cnd) R[rB] ← valE
PC update	PC ← valP

七、

1）

```
void count_pos1 (List *p, int *k) {
    int I, num=0, len;
    len = length(p)
    for (i = 0; i <len; i++) {
        if ( p->data    > 0)
            num++;
        p = p->next;
    }
    *k = num
}
```

2） while(p) 或其他相同功能的语句

3） L1 和 L2 的代码没有数据依赖，完全可以并行。
CPE 的下限为 3+1=4。

八、

1）

V	TAG	Block	V	TAG	Block
1	10	[8-9]	0		
1	01	[6-7]	1	10	[10-11]

2） 4

3）

V	TAG	Block	V	TAG	Block
1	10	[8-9]	0		
1	00	[2-3]	1	01	[6-7]

4） 4

2013 年

一、1. A

2. ABCD

3.D

4.C

5.D

6..D

7.ABC

8.ACD

9.BD

10.CD

11. ABBB

12 AAAB

13 AABC

14 ABD

15 AC

16 D

17 A

二、

1）

	True or false	原因或举出反例
if x<0, then x*2 <0	F	X= -2^w-1
u<= -1	T	-1 作为无符号来比大于 u
if x> y, then -x < -y	F	X=0, y= -2^w-1
if u> v, then -u > -v	F	U=2, v=1

2）

Value	$(-1)^s$ x M x 2^E 1<=M<2	Hex representation
$-1\frac{1}{2}$	(-1) x 1.1 x 2^0	0xBFC00000
2^{-149}	1.0 x 2^{-149}	0x00000001

三、

1、 答：64

2、

<int_sqrt>:

```
4004c4:        push   %rbp
4004c5:        mov    %rsp,%rbp
4004c8:        mov    %rdi,-0x28(%rbp)
4004cc:        movq   $0x0,-0x8(%rbp)
4004d4:        cmpq   $0x1,-0x28(%rbp)
4004d9:        ja     4004e1 <int_sqrt+0x1d>
```

```
4004db:         mov     -0x28(%rbp),%rax
4004df:         jmp     40052f <int_sqrt+0x6b>
4004e1:         movl    $0x0,-0x10(%rbp)
4004e8:         movl    $0x40000000,-0xc(%rbp)
4004ef:         jmp     400524 <int_sqrt+0x60>
4004f1:         mov     -0x10(%rbp),%rax
4004f5:         mov     -0x8(%rbp),%rdx
4004f9:         lea     (%rdx,%rax,1),%rax
4004fd:         mov     %rax,-0x18(%rbp)
400501:         shrq    -0x8(%rbp)
400505:         mov     -0x28(%rbp),%rax
400509:         cmp     -0x18(%rbp),%rax
40050d:         jb      40051f <int_sqrt+0x5b>
40050f:         mov     -0x18(%rbp),%rax
400513:         sub     %rax,-0x28(%rbp)
400517:         mov     -0x10(%rbp),%rax
40051b:         add     %rax,-0x8(%rbp)
40051f:         shrq    $0x2,-0x10(%rbp)
400524:         cmpq    $0x0,-0x10(%rbp)
400529:         jne     4004f1 <int_sqrt+0x2d>
40052b:         mov     -0x8(%rbp),%rax
40052f:         leaveq
400530:         retq
```

四、

1、答案：

```
#define N     3
#define M     5

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)
{
    static unsigned int x = sizeof(P1);
    static unsigned int y = sizeof(P2);

    if (n<=1)
        return 1;

    if ((n & 1) == 0)
        x++;

    if ((n & 1) == 1)
        y++;

    return f(n-1) + (y) +    (x);
}
```

2、答案：4f, 4e

五、1）1000/(280 + 20) = 1000/300 = 3.33GIPS

2）1000/(80 +60 +280 + 60 +20) = 1000/ 500 = 2 GIPS

3）

```
Prog：
irmovl    $128, %edx        ; 1、dx= 128
irmovl    $3, %ecx          ; 2、cx= 3
rmmovl    %ecx, 0(%edx)     ; 3、[dx] = cx
irmovl    $10, %ebx         ; 4、bx= 10
mrmovl    0(%edx), %eax     ; 5、ax= [dx]
addl      %ebx, %eax        ; 6、bx = bx + ax
```

相关：1-3，2-3, 1-5，4-6，5-6

冒险：1-3，2-3，4-6，5-6

4）2-3，5-6：执行到译码的转发通路解决；1-3,4-6：写回到译码的转发通路解决。

六、

Stage	cmovXX rA, rB	call Dest	decl rA
Fetch	icode:ifun ← M_1[PC] rA:rB ← M_1[PC+1] valP ← PC+2	icode:ifun ← M_1[PC] valC ← M_4[PC+1] valP ← PC+5	icode:ifun ← M_1[PC] rA:rB ← M_1[PC+1] valP ← PC+2
Decode	valA ← R[rA]	 valB ← R[%esp]	valA ← R[rA]
Execute	valE ← 0+valA Cnd ← Cond(CC,ifun) （也可以写Set CC）	valE ← valB+(-4)	valE ← valA+(-1) Cnd ← Cond(CC,ifun) （也可以写Set CC）
Memory	none	M_4[valE] ← valP	none
Write back	if(Cnd) R[rB] ← valE	R[%esp] ← valE	R[rA] ← valE
PC update	PC ← valP	PC ← valC	PC ← valP

七、

（1） {while(*src) *tgt++ = *src++ + delta; *tgt = 0;}

（2） 42

每次循环的关键路径为 读内存、做加法、写内存，该路径需要 42 个时钟周期。

本题陷阱：同学可能会受书上的例子误导认为做加法可以和下一个时钟周期的读内存并行，使得 CPE 降到 40，但实际上因为 src 和 tgt 指向的位置可能重叠，我们不能把下一次迭代的读操作移动到这一次的写操作之前。

（3） (((short) delta) << 8) + delta

*(src+1)

0xFF00
*src

八、1)

V	**TAG**	**DATA**
1	00	M[0-1]
1	00	M[2-3]
1	00	M[4-5]
0		

2)

V	TAG	DATA	V	TAG	DATA
1	000	M[0-1]	1	010	M[8-9]
1	000	M[2-3]	0		

6

3)

M[0-1] , M[8-9], M[16-17], M[2-3]