ICS第七次小班课

11.19

TA: 唐雯豪

内容安排

- 期中讲评
- Bomblab回课
- Archlab回课

选择题

第一题 单项选择题（每小题 2 分，共 24 分）

注：选择题的回答必须填写在下表中，写在题目后面或其他地方，视为无效。

题号	1	2	3	4	5	6
回答	A	C	A	C	B	A
题号	7	8	9	10	11	12
回答	C	B	B	A	D	B

选择题

- 3：`pa[0]` 是什么？
- 5：`0x24`是多少？
- 7：**SEQ需要预测跳转吗？** `rrmovq %rax, %rax` 会带来什么问题？
- 8：D选项有什么问题？
- 9：指令的空间局部性？
- 11：MRU在什么时候有用？
- ~~12：SDRAM为什么不是SRAM和DRAM融合成的~~

例子：全相联cache大小为3字节，访存顺序1 2 3 4 1 2 3 4，LRU全miss，MRU hit三次

关于缓存策略的问题，一般来说，A策略比B策略好，当且仅当B策略hit的时候A策略一定hit。（不一定正确，还是要分析具体问题）

第二题

- 最大的负非规格化数？
  - -0.0也算对
- 注意小端法
- 注意两个空别填反了

（1）以上程序的两次输出分别是 ____NaN____（2 分），____1____（2 分）。

（2） sizeof(union_e) = ____8____。（2 分）

（3）如果某次程序输出的 test.d 是最大的负非规格化数，则

此时从 test 的起始地址开始的 8 个字节（用 16 进制表示）依次是：（4 分，全对才得分，否则 0 分）

01	00	00	00	00	00	00	80

此时

（1） 这段代码的运行输出结果是（若有多个输出结果，用逗号分隔表示）：

_____ -2，1，0，-1，2_____（3 分，全对才得分）

（2）进一步分析，float 类型不可以精确表示的最小正整数是 ___ 2**24+1___。（3 分）

test.s.x = _____ 1_____（2 分）

test.s.y = ___-2147483648 (-2**31)___（2 分）

第三题

参考答案：

1、（注：本题十六进制未带 0x，不扣分）

（1）return;　(注：未带分号，不扣分)

（2）params->n

（3）%rdi

（4）-0xc

（5）%rbp

（6）sub　(注：写成 subq，不扣分)

（7）jle

（8）%rax

（9）%edx

（10）-0x8(%rbp)

（11）0x00005555555551af＜foo＞　（只写数值或者<foo>，都算正确）

（12）0x555551f9

（13）0x555551de

（14）0xffffe2f0

2、计算阶乘

第三题

- 1.(8)填 %rax,%eax 都对
  - 课本上说只能基址寄存器和变址寄存器只能是64位，实际编译器支持lea指令使用32位

```
1 #include<stdio.h>
2 void main(){
3   __asm("lea -0x1(%eax), %edx");
4   __asm("lea -0x1(%eax), %rdx");
5   __asm("lea -0x1(%rax), %edx");
6   __asm("leaq -0x1(%rax), %edx");
7   __asm("leaq -0x1(%rax), %rdx");
8 }
```

```
root@AY130903143224977a6fZ:~# gcc tt.c
tt.c: Assembler messages:
tt.c:6: Error: incorrect register `%edx' used with `q' suffix
```

In 64-bit mode, the instruction's destination operand is governed by operand size attribute, the default operand size is 32 bits. Address calculation is governed by address size attribute, the default address size is 64-bits. In 64-bit mode, address size of 16 bits is not encodable. See Table 3-55.

Operand Size	Address Size	Action Performed
16	32	32-bit effective address is calculated (using 67H prefix). The lower 16 bits of the address are stored in the requested 16-bit register destination (using 66H prefix).
16	64	64-bit effective address is calculated (default address size). The lower 16 bits of the address are stored in the requested 16-bit register destination (using 66H prefix).
32	32	32-bit effective address is calculated (using 67H prefix) and stored in the requested 32-bit register destination.
32	64	64-bit effective address is calculated (default address size) and the lower 32 bits of the address are stored in the requested 32-bit register destination.
64	32	32-bit effective address is calculated (using 67H prefix), zero-extended to 64-bits, and stored in the requested 64-bit register destination (using REX.W).
64	64	64-bit effective address is calculated (default address size) and all 64-bits of the address are stored in the requested 64-bit register destination (using REX.W).

Table 3-55. 64-bit Mode LEA Operation with Address and Operand Size Attributes

第三题

- 2.注意%rbp和返回地址都是64位的

请填写空格中对应的值。（每空 2 分，共 6 分）

地址	值
0x7fffffffe308 foo	0xffffe340
0x7fffffffe304	0x00000000
0x7fffffffe300	0x00000000
0x7fffffffe2fc RA	0x00005555
0x7fffffffe2f8	(12) 0x5555 51f9
0x7fffffffe2f4 foo %rbp	0x00007fff
0x7fffffffe2f0	0xffffe310
0x7fffffffe2ec mov %rsp, %rbp	0x00007fff
0x7fffffffe2e8 %rsp -0x10	0xffffe340
0x7fffffffe2e4	0x00000004
0x7fffffffe2e0	0xffffe350
0x7fffffffe2dc RA	0x00005555
0x7fffffffe2d8	(13) 0x5555 51de
0x7fffffffe2d4 bar %rbp	0x00007fff
0x7fffffffe2d0	(14) 0xffff e2f0

第四题

- 第一问
- 注意SEQ的实现带来的格式问题（不能随意修改SEQ的实现结构）：
  - +8只能是valB
  - 访存只能用valA（或valE）
  - rA, rB应该是取指阶段拿出来的，随便使用不太合适
- 只读valA是否扣分？

取指	icode:ifun ← M_1[PC] valP ← PC + 1
译码	（每行操作2分，一行全对才得分。共2行，合计4分） **valA ← R[%rbp]** **valB ← R[%rbp]**
执行	（4分，一行全对才得分） **valE ← valB + 8**
访存	（4分，一行全对才得分） **valM ← M_8[valA]**
写回	（每行操作2分，一行全对才得分。共2行，合计4分） **R[%rsp] ← ValE** **R[%rbp] ← ValM**
更新PC	PC ← valP

第四题

（2）类似的，如果尝试新加入指令enter。其语义相当于

```
pushq       %rbp
rrmovq      %rsp, %rbp
```

为执行拟新增的指令，你尝试改进PIPE处理器，最终完成的工作为（可多选）：

含有C但不含A不含B的组合，或单选G 。

补充说明：选G是常规思路；但是，含C的选项并试图维持可能的流水控制信号的设计也可以算对。流水线处理器中里面的寄存器堆在常规情况下只安排2个写口。含C选项相当于安排3个写口，虽然不是常规设计但仍可以实现。

第五题

- (1) 3次
- (2)
  - 1次，给2分
  - 5次，给1分（认为访问两个字节是两次单独的访存操作）
  - 注意是非写分配
  - 别忘了填那个图

R 0 [000000$_2$], ✗
R 1 [000001$_2$], ✓
R 7 [000111$_2$], ✗
R 4 [000100$_2$], ✓
R 0 [000000$_2$] ✓

Set 2

v	Tag	Block
1	00	M[8-11]

R 0 [000000$_2$], ✗
R 1 [000001$_2$], ✓
W 7 [000111$_2$], ✗ ✗
R 12 [001100$_2$], ✗
R 9 [001001$_2$] ✗

Set 1

v	Tag	Block
0		

第五题

- (3)
  - 10或者12，给4分
    - 12: 考虑每个miss要带一次hit
  - 18或者20，给2分（认为访问两个字节是两次单独的访存操作）