南京航空航天大学《计算机组成原理Ⅱ课程设计》报告

- 姓名：唐希
- 班级：1619201
- 学号：161920122
- 报告阶段：PA2.2-2.3
- 完成日期：2021.5.25
- 本次实验，我完成了几乎所有内容。

目录

(2)add-longlong.c
(3)bit.c
6a(push)
c1(sar)
d3(shl)
22(and)
84(test)
0f 95(set)
0a(or)
21(and)
(4)bubble-sort.c
2b(sub)
40(inc)
(5)fact.c
(6)fib.c
48(dec)
(7)goldbach.c
(8)hello-str.c
68(push)
0f be(movsx)
ff(jmp)
43(inc)
29(sub)
(9)if-else.c
(10)leap-year.c
05(add)
(11)load-store.c
0f bf(movsx)
0f b7(movzx)
(12)matrix-mul.c
0f af(imul)
(13)max.c
(14)min3.c
(15) mov-c.c
(16)movsx.c
(17) mul-longlong.c
0b(or)
(18)pascal.c
(19)prime.c
(20)quick-sort.c
ff(dec)
(21)recursion.c
ff(call)
c1(shr)
(22)select-sort.c
(23)shift.c
(24)shuixianhua.c
(25)string.c
(26)sub-longlong.c
1b(sbb)
(27)sum.c
(28)switch.c
(29)to-lower-case.c
(30)unalign.c
(31) wanshu.c
2.通过一键回归测试
3.IN/OUT 指令
加入IOE

思考题和git

1.什么是API?

API即Application Programming Interface，译为应用程序接口。是一种计算接口，它定义多个软件中介之间的交互，以及可以进行的调用（call）或请求（request）的种类，如何进行调用或发出请求，应使用的数据格式，应遵循的惯例等。它还可以提供扩展机制，以便用户可以通过各种方式对现有功能进行不同程度的扩展。一个API可以是完全定制的，针对某个组件的，也可以是基于行业标准设计的以确保互操作性。通过信息隐藏，API实现了模块化编程，从而允许用户实现独立地使用接口。

2.AM属于软件还是硬件?

AM 是抽象计算机类似于操作系统对于一般计算机的概念一样，通过一组统一的API实现对计算机底层细节的抽象，为程序运行提供最基本的软件支持，是连接硬件与软件的桥梁。

操作系统是位于硬件与软件之间，而AM也是位于NEMU（模拟硬件）与软件之间，他们的功能都是实现对硬件的抽象，都是通过API实现的，我认为他们的功能是一样的

3.堆和栈在哪里?

因为堆和栈都需要频繁进行出栈和入栈等数据操作，所以被放在内存里面，在执行数据操作的时候动态地申请和清空。

4.回忆运行过程

```
make ARCH=x86-nemu ALL=dummy run
```

命令在make run之中插入了两段命令

一、根据命令中的 `ARCH=x86-nemu` 可以知道，程序被默认编译到了x86-nemu之中

二、根据命令中的 `ALL=dummy` 可以知道，整个命令通过调用 `nexus-am/am/arch/x86-nemu/img/run` 目录来启动NEMU。

三、根据主命令make run，最终运行dummy.c程序。

5.神奇的eflags （2）

SF	OF	实例
0	0	2 - 1
0	1	0xf0000000-0x00000001
1	0	0x80000001-0x00000001
1	1	0x0f000000-0x00000001

6.这是巧合吗?

1、ja

无符号整数op1>op2

CF=0 AND ZF=0

2、jb

无符号整数op1<op2

CF=1 AND ZF=0

3、jg

带符号整数op1>op2

SF=OF AND ZF=0

4、jl

带符号整数op1<op2

SF≠OF

7. NEMU的本质

两个整数相加的小程序

```
label1:
   x=x-1
       a=a+1
       jne x,label1

label2:
   y=y-1
       a=a+1
       jne y,label2
```

我认为NEMU还需要：

和windows类似的图形操作界拆第面；

集成化的调试和运行编译器；

输入和输出；

图片和视频；

8.设备是怎么工作的?

通过I/O接口实现CPU和外部设备的信息交换，CPU通过一系列的发送端口向相应的设备发出指令信号，设备接收之后执行自己的工作，执行结束或者遇到错误发送反馈信号通过接口传给CPU。

9. CPU 需要知道设备是如何工作的吗?

不需要，CPU只要负责传递指令和数据给设备，然后等待设备传递回相应的数据即可。

10. 什么是驱动?

驱动程序即添加到操作系统中的一小块代码，一般是硬件厂商根据操作系统编写的配置文件。其中包含有关硬件设备的信息。有了此信息，计算机就可以与设备进行通信。

通俗的讲：各个电脑硬件需要一个中间件来连接，驱动的作用就是让硬件和操作系统进行通信，把硬件的操作转换成机器语言。

11. cpu知道吗?

不需要，只需要把指定地址上的值修改为指定的值就好了。

12.再次理解volatile

如果 0x8048000 被映射到一个设备寄存器将检测不到设备寄存器的变化从而进入死循环

13.hello world运行在哪里?

不一样。程序设计课的Hello World 程序运行在硬件层，而我们这个hello程序运行在AM层

14.如何检测很多个键同时被按下?

当按下键盘的一个按键时，键盘会向CPU发送相应的键盘码，这些键盘码会放入数据寄存器， 同时把状态寄存器的标志设置为 `1`， CPU 通过端口 I/O 访问这些寄存器，通过标志寄存器的值来判断按键，进而实现相应的操作。

15.编译与链接 I

去掉了很多`static`但是运行`nemu`的时候并没有报错。

去掉`inline`以后出现定义但未使用错误，`inline`用来把一些频繁使用的函数放到栈里面提高运行效率。去掉这个关键字后这个函数就没有定义在栈区，只能在栈区内使用 ，其他函数调用时在栈区外寻找找不到，所以这个函数定义了但是没有被使用，才会报错。

去掉了`static`和`inline`后出现重复定义错误，当多个文件包含同一个头文件时，而头文件中没有加上条件编译，会形成独立解释，在编译器链接生成.o文件的时候就会出现重复定义错误。

16.编译与链接Ⅱ

1、重新编译后的 NEMU 含有多少个 dummy 变量的实体有29个。可以通过grep -rn "dummy" |wc -l指令来查看此时的 NEMU 含有多少个 dummy 变量的实体

2、有58个，多了29个，加上了debug.h里面的。 同样通过上面那条指令查看。

3、会有一个连接错误，他没初始化之前是若符号，所以不会报错，根据他之前定义的强符号处来处理，而定义　　了强符号也就是初始化之后呢就会出现两个强符号大家的情况，自然就是连接错误。

17.I/O 端口与接口

因为1K=0400H，所以系统 I/O 地址的范围是 0000H ~ 03FFH ； 166根地址总线，寻址范围也 就是2^16，因为1K=2^10，所以寻址范围为2^16/2^10=64K，所以端口的地址范围是 0000H ~ FFFFH

I/O三种控制方式：程序直接控制、终端控制、DMA控制。 首先DMA控制器初始化，然后发送“启动DMA传送”命令以启动外设进行I/O操作，发送完“启 动DMA传送”命令后， CPU转去执行其他进程，而请求I/O的用户进程被阻塞。在CPU执行其 他进程的过程中，DMA控制器外设和主存进行数据交换。DMA控制器每完成一个数据的传送，就将字计数器减1，并修改主存地址，当字计数器为0时，完成所有I/O操作，此时，DMA 控制器将向CPU发送“DMA完成”中断请求，CPU检测到后调出相应的中断服务程序执行。CPU 在中断服务程序中，解除用户进程的阻塞状态而使用户进程进入就绪序列，然后中断返回，再回到被打断的进程继续执行。

例如，磁盘与内存间的信息交换，希望用硬件在外设与内存间直接进行数据交换,而不通过CPU,这样数据传送的速度的上限就取决于存储器的工作速度.但是,通常系统的地址和数据总线以及一些控制信号线(例如I/O等)是由CPU管理的，所以当CPU发出DMA响应信号之后,DMA控制器接管对总线的控制.在DMA方式时,CPU把这些总线让出来，由他接管,控制传送的字节数,判断DMA是否结束,以及发出DMA结束等信号.这些都是由硬件实现的.在DMA传送结束以后,他结束DMA请求信号,释放总线,使CPU恢复正常工作。

18. git log截图

```
tangxi@debian: ~/ics2021
commit a9562efb1daf4e650f0508f25d00c9b4ece03236 (HEAD -> pa2)
Author: tracer-ics2017 <tracer@njuics.org>
Date:   Mon Jun 7 10:09:52 2021 +0800

    > run
    161920122
    tangxi
    Linux debian 4.19.0-14-686 #1 SMP Debian 4.19.171-2 (2021-01-30) i686 GNU/Linux
     10:09:52 up 1 day, 49 min,  1 user,  load average: 1.28, 0.53, 0.27
    ad7d1c2a098e9aa7b47ca289f1ff09092348961

commit 19c7fdaad3239b47524256221cfb92cfd99190da
Author: tracer-ics2017 <tracer@njuics.org>
Date:   Mon Jun 7 10:07:38 2021 +0800

    > run
    161920122
    tangxi
    Linux debian 4.19.0-14-686 #1 SMP Debian 4.19.171-2 (2021-01-30) i686 GNU/Linux
     10:07:38 up 1 day, 47 min,  1 user,  load average: 0.00, 0.00, 0.08
    9115870dcc0890d9c011fe89a913e5a35e2354ee

commit 9f10d16393468844d7432c15de84af16e6645820
Author: tracer-ics2017 <tracer@njuics.org>
Date:   Mon Jun 7 09:27:43 2021 +0800

    > run
    161920122
    tangxi
    Linux debian 4.19.0-14-686 #1 SMP Debian 4.19.171-2 (2021-01-30) i686 GNU/Linux
     09:27:43 up 1 day, 7 min,  1 user,  load average: 0.28, 0.72, 0.54
    8694b01f3aef674516500a22b64cd95111b3994d

commit 25be30bf83e686e92fa26863bba74fb7e9b203b0
Author: tracer-ics2017 <tracer@njuics.org>
Date:   Mon Jun 7 09:21:27 2021 +0800

    > run
    161920122
    tangxi
    Linux debian 4.19.0-14-686 #1 SMP Debian 4.19.171-2 (2021-01-30) i686 GNU/Linux
     09:21:27 up 1 day, 1 min,  1 user,  load average: 0.00, 0.08, 0.24
    78a72375e692842bf6bfeada2af2564df09af61

commit 28efa13ac02be74a3a2249e1f8c98b34dd19d2ed
Author: tracer-ics2017 <tracer@njuics.org>
Date:   Mon Jun 7 09:08:42 2021 +0800

    > run
    161920122
    tangxi
:
```

```
tangxi@debian:~/ics2021$ git push myrepo pa2
Username for 'https://e.coding.net': 15913121302
Password for 'https://15913121302@e.coding.net':
Enumerating objects: 961, done.
Counting objects: 100% (961/961), done.
Compressing objects: 100% (925/925), done.
Writing objects: 100% (927/927), 88.88 KiB | 1.06 MiB/s, done.
Total 927 (delta 844), reused 1 (delta 0)
remote: Resolving deltas: 100% (844/844), completed with 23 local objects.
To https://e.coding.net/tangxi1/tangxi/PA.git
   8a8ab54..a9562ef  pa2 -> pa2
tangxi@debian:~/ics2021$
```

操作题

PA2.2基本指令集

1.实现剩余所有 x86 指令

约有 40 个（有的只需要填表），以跑通所有测试用例为准；
指令实现的顺序为：
运行一个新的测试用例；
该测试用例新出现了某些指令；
逐个实现每个出现的指令；
本测试样例成功运行，开始下一个测试用例直至全部用例通过

进入`ics2021/nexus-am/tests/cputest/tests`目录下，查看所有的测试样例

```
tangxi@debian:~/ics2021/nexus-am/tests/cputest/tests$ ls
add.c            fib.c          matrix-mul.c     pascal.c        shuixianhua.c    unalign.c
add-longlong.c   goldbach.c     max.c            prime.c         string.c         wanshu.c
bit.c            hello-str.c    min3.c           quick-sort.c    sub-longlong.c
bubble-sort.c    if-else.c      mov-c.c          recursion.c     sum.c
dummy.c          leap-year.c    movsx.c          select-sort.c   switch.c
fact.c           load-store.c   mul-longlong.c   shift.c         to-lower-case.c
tangxi@debian:~/ics2021/nexus-am/tests/cputest/tests$
```

在 `nexus-am/tests/cputest/` 目录下执行

make all=add.c run

(1)add.c

8d(lea)

```
10005e:   8d 8d 4c 24 04 83 e4 f0 ff              invalid opcode
```

8d这个地址没有实现，查表发现是

LEA

Gv,M

LEA指令，译码函数是

lea_M2G，执行函数是lea，去填表

```
/* 0x8c */    EMPTY, IDEX(lea_M2G, lea), EMPTY, EMPTY,
```

去`all-instr.h`声明lea,`make_EHelper(lea);`

注意：后续所有新添加的`make_EHelper(xxx);`都要去`all-instr.h`里面声明，不再做重复说明

```
  100029:   e8 30 00 00 00                      call 10005e
  10005e:   8d 4c 24 04                         leal 0x4(%esp),%ecx
invalid opcode(eip = 0x00100062): 83 e4 f0 ff 71 fc 55 89 ...
```

运行成功

83(and)

这个地址没有实现，查看反汇编知道是and没有实现，查表发现第五个是and

p	000	001	010	011	100
1	ADD	OR	ADC	SBB	AND

填表

```
make_group(gp1,
    EMPTY,  EMPTY,  EMPTY,  EMPTY,
    EX(and),  EX(sub),  EMPTY,  EMPTY)
```

同时得知译码函数是SI2E，去修改译码函数SI，实现位拓展

```
static inline make_DopHelper(SI) {
   assert(op->width == 1 || op->width == 4);

   op->type = OP_TYPE_IMM;

   /* TODO: Use instr_fetch() to read `op->width' bytes of memory
    * pointed by `eip'. Interpret the result as a signed immediate,
    * and assign it to op->simm.
    *
    op->simm = ???
    */
   op->simm=instr_fetch(eip,op->width);

   rtl_li(&op->val, op->simm);

   rtl_sext(&op->val, &op->val, op->width);//截断和符号扩展

   op->simm=op->val;//更新simm

 #ifdef DEBUG
   snprintf(op->str, OP_STR_SIZE, "$0x%x", op->simm);
 #endif
 }
```

在`logic.c`中完成and函数

```
make_EHelper(and) {
  rtl_and(&id_dest->val, &id_dest->val, &id_src->val);//目的操作数与源操作数相与
  operand_write(id_dest, &id_dest->val);//写入目的操作数
  t0=0;
  rtl_update_ZFSF(&id_dest->val, id_dest->width);//更新ZFSF位
  rtl_set_CF(&t0);//设置CF位为0
  rtl_set_OF(&t0);//设置OF位为0
  print_asm_template2(and);
}
```

```
100062:   83 e4 f0                          andl $0xfffffff0,%esp
```

运行成功

eb(jmp)

```
invalid opcode(eip = 0x0010007d): eb 5d c7 45 e8 01 00 00 ...
```

eb没有实现，查表

JNP		
Jv	Ap	Jb
STC	CLI	STI

译码函数选择J，执行函数选择jmp操作数长度位1字节。

填表

```
/* 0xe8 */  IDEXW(J, call, 4),  EMPTY,  EMPTY,  IDEXW(J, jmp, 1)
```

运行成功截图

```
invalid opcode(eip = 0x001000dc): 81 7d f4 96 00 00 00 7e ...
```

83(cmp)

查表可以知道83是cmp指令，根据指令含义只需要前一个数减去后一个数就行，根据结果判断就行。

```
make_EHelper(cmp) {

 rtl_sub(&t2, &id_dest->val, &id_src->val);
 rtl_sltu(&t3, &id_dest->val, &t2);//t3记录是否借位，0表示借位

 rtl_update_ZFSF(&t2, id_dest->width);//更新ZF，SF

 rtl_sltu(&t0, &id_dest->val,  &t2);//与减去借位后再比
 rtl_or(&t0, &t3, &t0);
 rtl_set_CF(&t0);

 rtl_xor(&t0,  &id_dest->val, &id_src->val);
 rtl_xor(&t1, &id_dest->val, &t2);
 rtl_and(&t0, &t0, &t1);
 rtl_msb(&t0, &t0, id_dest->width);
 rtl_set_OF(&t0);

 print_asm_template2(cmp);
 }
```

填表

```
make_group(gp1,     EMPTY,  EMPTY,  EX(adc),  EMPTY,     EX(and),  EX(sub),
EMPTY,  EX(cmp))
```

运行成功截图

```
100065:   ff 71 fc                                pushl -0x4(%ecx)
100068:   55                                      pushl %ebp
100069:   89 e5                                   movl %esp,%ebp
10006b:   51                                      pushl %ecx
10006c:   83 ec 14                                subl $0x14,%esp
10006f:   c7 45 ec 00 00 00 00                    movl $0x0,-0x14(%ebp)
100076:   c7 45 f4 65 00 00 00                    movl $0x65,-0xc(%ebp)
10007d:   eb 5d                                   jmp 1000dc
1000dc:   81 7d f4 96 00 00 00                    cmpl $0x96,-0xc(%ebp)
```

这里开始反汇编就与nemu不一样了

0f 94(set)

查看`0x0f`处和反汇编发现是`set`操作

`100112: 0f 94 c0  sete  %al`

填表

```
/* 0x94 */  IDEXW(E, setcc, 1), EMPTY, EMPTY, EMPTY,
```

在`logic.c`文件中找到了相应的执行函数

```
make_EHelper(setcc) {
  uint8_t subcode = decoding.opcode & 0xf;
  rtl_setcc(&t2, subcode);
  operand_write(id_dest, &t2);

  print_asm("set%s %s", get_cc_name(subcode), id_dest->str);
}
```

发现`rtl_setcc`函数没有被实现。

需要实现`rtl_setcc`

在`cc.c`文件里面找到了`rtl_setcc`函数

查询`i386`，可知

对照要求，依次实现不同`cc`对`eflags`的读取即可

```
 switch (subcode & 0xe) {
    case CC_O://0
      rtl_get_OF(dest);
      break;
    case CC_B://2
      rtl_get_CF(dest); //小于，通过CF来判断不够减
      break;
    case CC_E://4
```

```
        rtl_get_ZF(dest);
        break;
      case CC_BE: { //6
        rtl_get_CF(&t0);
        rtl_get_ZF(&t1);
        rtl_or(dest, &t0, &t1); //小于等于，CF和ZF至少一个要等于1才行
      }break;
      case CC_S: //8
        rtl_get_SF(dest);
        break;
      case CC_L: { //12
        rtl_get_SF(&t0);
        rtl_get_OF(&t1);
        rtl_xor(dest, &t1, &t0); //带符号数的小于，SF不能等于OF
      }break;
      case CC_LE: { //14
        rtl_get_ZF(&t0);
        rtl_get_SF(&t1);
        rtl_get_OF(&t2);
        rtl_xor(&t3, &t1, &t2);
        rtl_or(dest, &t0, &t3); //带符号数的小于等于，ZF=1或者SF不等于OF
      }break;
      default: panic("should not reach here");
      case CC_P: panic("n86 does not have PF");
    }

    if (invert) {
      rtl_xori(dest, dest, 0x1);
    }
  }
```

运行成功截图

忘记截图了。。。

76(jbe)

译码函数`J`

执行函数`jcc`

填表`/* 0x74 */  EMPTY,  EMPTY,  IDEXW(J, jcc, 1),  EMPTY,`

运行成功截图

3b(cmp)

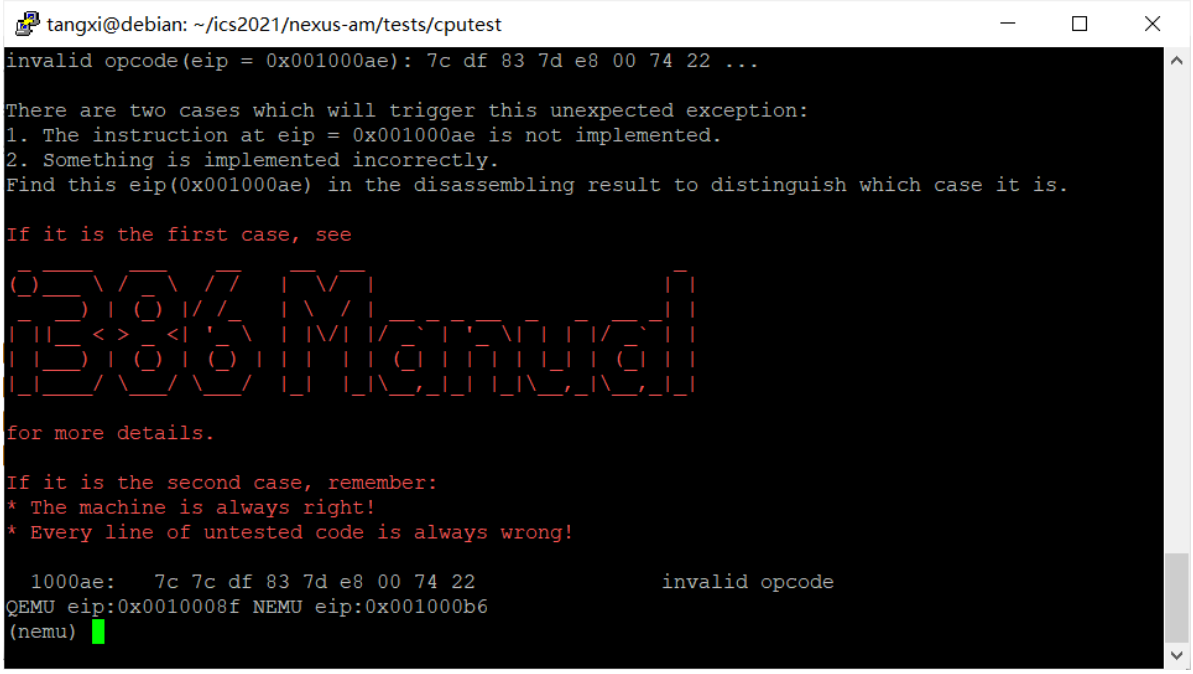
```
invalid opcode(eip = 0x001000ab): 3b 45 f4 7c df 83 7d e8 ...
```

P
Gv,Ev

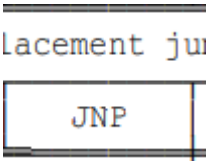
还是cmp指令，译码函数I2a,执行函数cmp，之前实现过了，填表

```
/* 0x38 */  IDEXW(G2E,cmp,1), IDEX(G2E,cmp), IDEXW(E2G,cmp,1), IDEX(E2G, cmp),
/* 0x3c */  IDEXW(I2a,cmp,1), IDEX(I2a,cmp), EMPTY, EMPTY,
```

运行成功截图



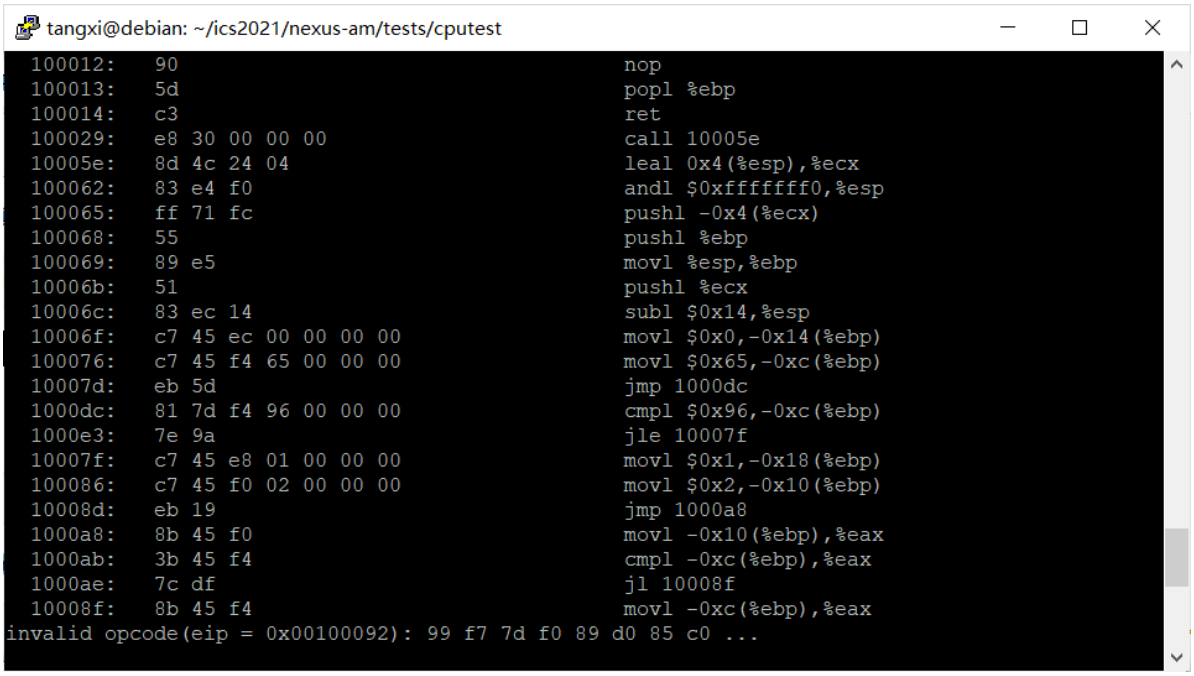
7c(jnp)



是jnp指令，译码函数J，执行函数jcc，宽度为1，填表

```
/* 0x7c */ IDEXW(J, jcc, 1), EMPTY, IDEXW(J, jcc, 1), EMPTY,
```

运行成功截图



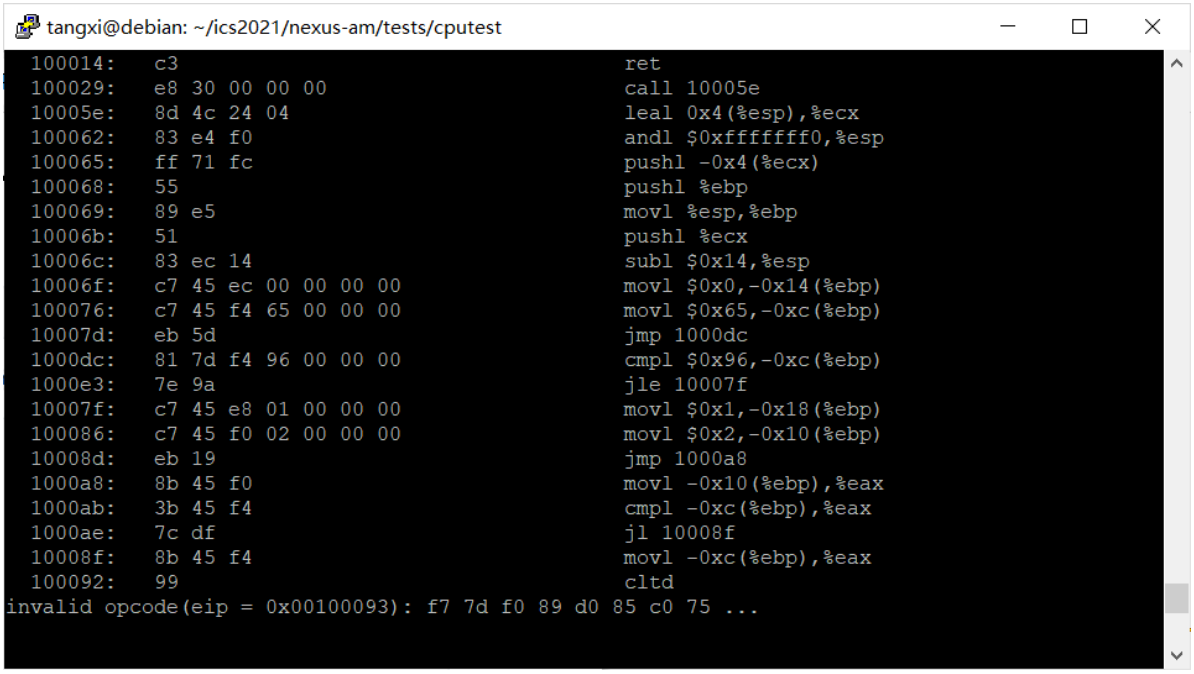
99(cltd)

99(cltd) 无译码函数，执行函数 cltd ， 填表 在 data-mov.c 文件中完成 cltd 函数，在 all-instr.h 中声明 make_EHelper(cltd)

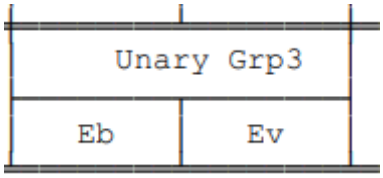
```
/* 0x98 */    EMPTY, EX(cltd), EMPTY, EMPTY,
```

```
make_EHelper(cltd) {
  if (decoding.is_operand_size_16) {
    rtl_lr_w(&t0, R_AX);
    rtl_sext(&t0, &t0, 2);
    rtl_sari(&t0, &t0, 16);
    rtl_sr_w(R_DX, &t0);

  }
  else {
   rtl_lr_l(&t0, R_EAX);
   rtl_sari(&t0, &t0, 31);
   rtl_sari(&t0, &t0, 1);
   rtl_sr_l(R_EDX, &t0);
  }

  print_asm(decoding.is_operand_size_16 ? "cwtl" : "cltd");
}
```

运行成功截图

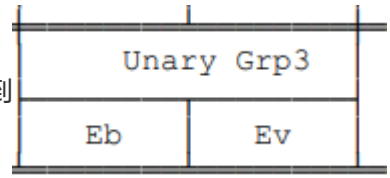


f7(test,not,neg,mul,imul,div,idiv)



反汇编代码不知道为何看不了，为了通过就把表全部填了

查表，得到



填表,声明

```
make_group(gp3,
    IDEX(test_I,test), EMPTY, EX(not), EX(neg),
    EX(mul), EX(imul1), EX(div), EX(idiv))
```

```
 /* 0xf4 */ EMPTY, EMPTY, IDEXW(E, gp3, 1), IDEX(E, gp3),
```

test

译码函数选择grp3里的test_I,执行函数test

先去`logic.c`里面实现test

```
make_EHelper(test) {
  rtl_and(&t0, &id_dest->val, &id_src->val);
  rtl_update_ZFSF(&t0, id_dest->width);
  rtl_set_CF(&tzero);
  rtl_set_OF(&tzero);

  print_asm_template2(test);
}
```

然后填表即可

```
make_group(gp3,
    IDEX(test_I,test), EMPTY, EMPTY, EMPTY,
    EMPTY, EMPTY, EMPTY, EMPTY)
```

not

在 logic.c 文件中完成 not 函数，在 all-instr.h 中声明 make_EHelper(not)

```
make_EHelper(not) {
  rtl_not(&id_dest->val);//取反
  operand_write(id_dest,&id_dest->val);
print_asm_template1(not);
}
```

填表

```
make_group(gp3,
    IDEX(test_I,test), EMPTY, EX(not), EMPTY,
    EMPTY, EMPTY, EMPTY, EMPTY)
```

neg

查手册，填表

```
make_group(gp3,
    IDEX(test_I,test), EMPTY, EX(not), EX(neg),
    EMPTY, EMPTY, EMPTY, EMPTY)
```

在 `arith.c` 中完成 `make_EHelper(neg)` ，

```
make_EHelper(neg) {
  if(!id_dest->val){
    rtl_set_CF(&tzero);
  }
  else{
    rtl_addi(&t0, &tzero, 1);
    rtl_set_CF(&t0);
  }
  rtl_add(&t0, &tzero, &id_dest->val);
  t0 = -t0;
  operand_write(id_dest, &t0);
  rtl_update_ZFSF(&t2, id_dest->width);
  rtl_xor(&t0, &id_dest->val, &id_src->val);
  rtl_xor(&t1, &id_dest->val, &t2);
  rtl_and(&t0, &t0, &t1);
  rtl_msb(&t0,&t0, id_dest->width);
  rtl_set_OF(&t0);

  print_asm_template1(neg);
}
```

mul

查表，无译码函数，执行函数 mul ， mul 已实现，在 `all-instr.h` 中声明 `make_EHelper(mul)` ，查阅 i386手册附录A可知需填写第 100 项

```
make_group(gp3,
    IDEX(test_I,test), EMPTY, EX(not), EX(neg),
    EX(mul), EMPTY, EMPTY, EMPTY)
```

imul

查表，无译码函数，执行函数 imul ， imul 已实现，在 `all-instr.h` 中声明 `make_EHelper(imul)` ，查阅 i386手册附录A可知需填写第 101 项

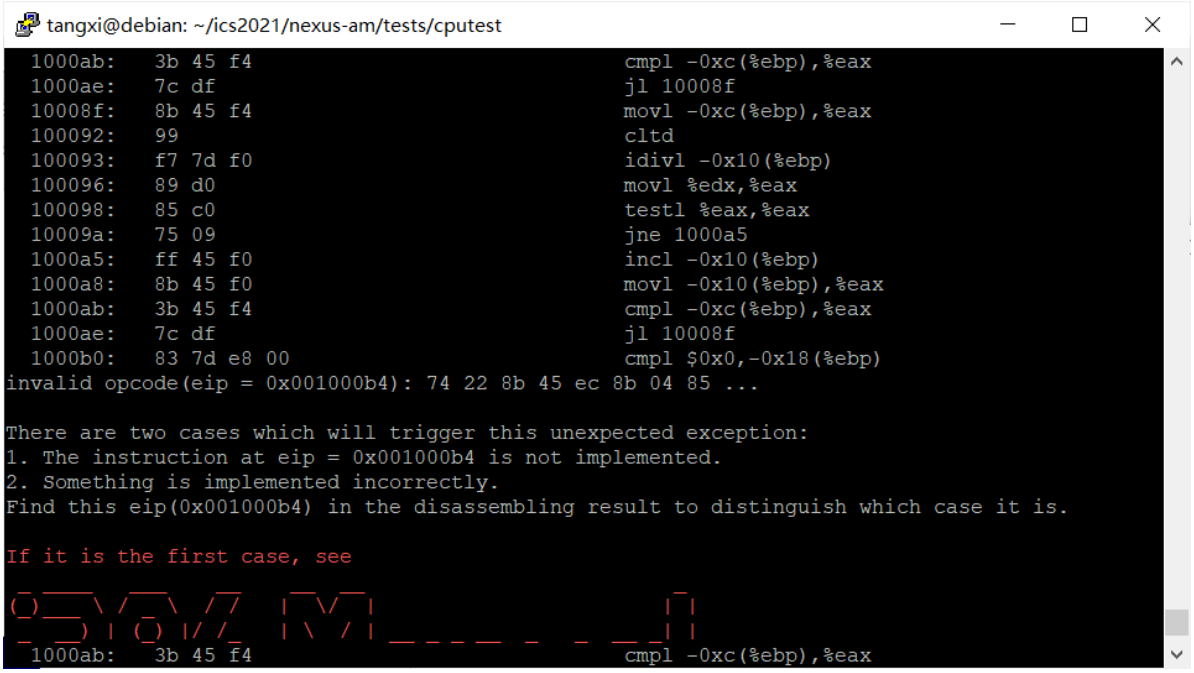
```
make_group(gp3,
    IDEX(test_I,test), EMPTY, EX(not), EX(neg),
    EX(mul), EX(imul), EMPTY, EMPTY)
```

div

div和idiv都已经实现了，查看手册分别 填写第110和111项即可，

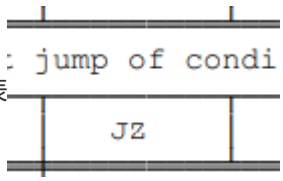
```
make_group(gp3,
    IDEX(test_I,test), EMPTY, EX(not), EX(neg),
    EX(mul), EX(imul1), EX(div), EX(idiv))
```

f7全部写完的运行成功截图



74(je)

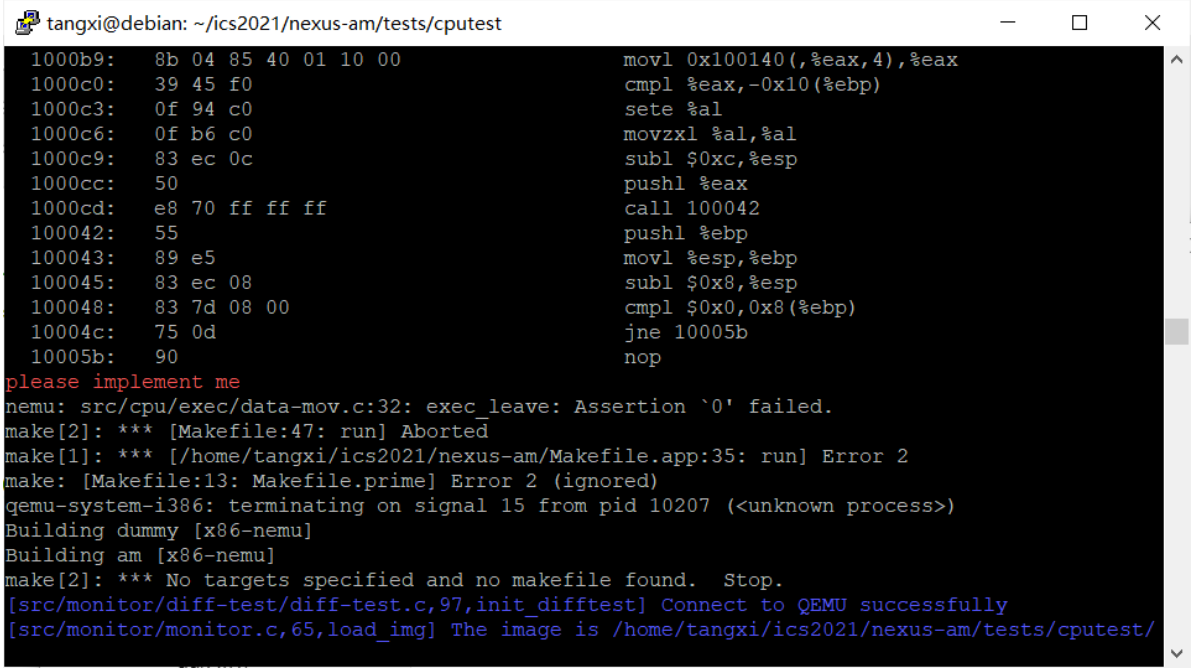
查表



译码函数J， 执行函数 jcc ， 宽度为1， 将70-7f全填了

```
/* 0x70 */    IDEXW(J, jcc, 1), IDEXW(J, jcc, 1), IDEXW(J, jcc, 1), IDEXW(J,jcc, 1),
  /* 0x74 */    IDEXW(J, jcc, 1), IDEXW(J, jcc, 1), IDEXW(J, jcc, 1), IDEXW(J,jcc, 1),
  /* 0x78 */    IDEXW(J, jcc, 1), IDEXW(J, jcc, 1), IDEXW(J, jcc, 1), IDEXW(J,jcc, 1),
  /* 0x7c */    IDEXW(J, jcc, 1), IDEXW(J, jcc, 1), IDEXW(J, jcc, 1), IDEXW(J,jcc, 1),
```

运行成功截图



c9(leave)

无译码函数，执行函数 leave 在 `data-mov.c` 文件中完成函数 leave

```
make_EHelper(leave) {
rtl_mv(&cpu.esp, &cpu.ebp);
rtl_pop(&cpu.ebp);
print_asm("leave");
}
```

填表

```
/* 0xc8 */ EMPTY, EX(leave), EMPTY, EMPTY,
```

运行成功截图

83(add,or)

又遇到了83，这次没办法通过反汇编来看指令了，将grp1填完吧

```
make_group(gp1,
EX(add), EX(or), EMPTY, EMPTY,
EX(and), EX(sub), EX(xor), EX(cmp))
```

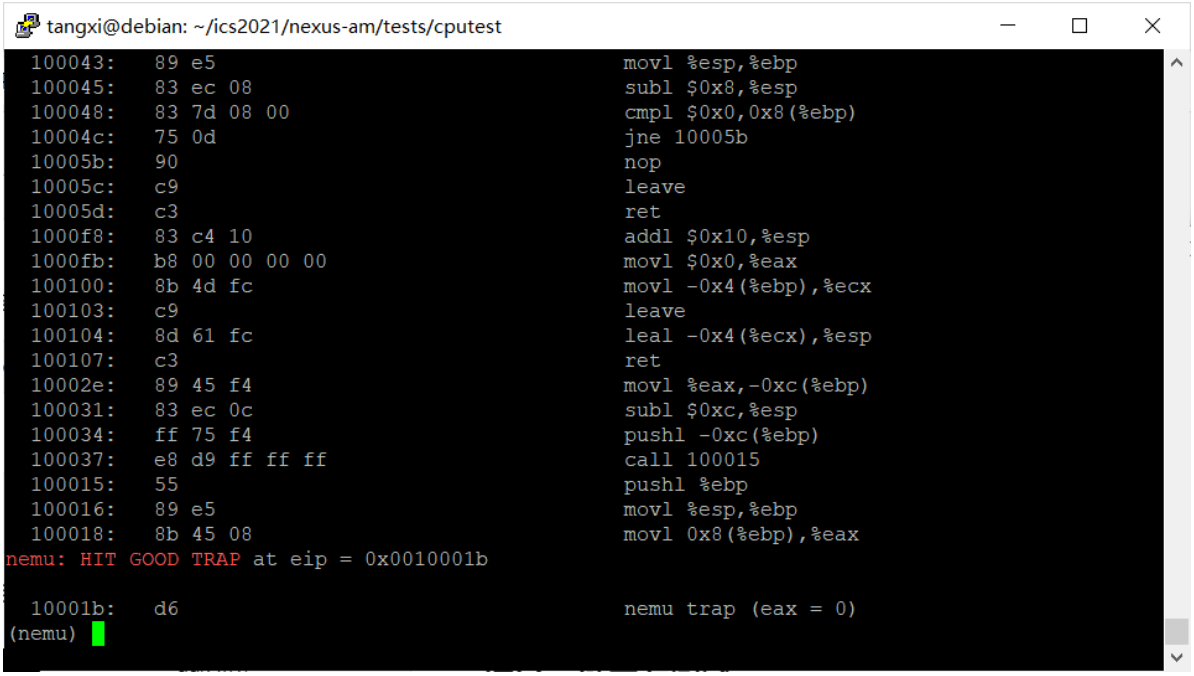
先实现add

在 `arith.c` 文件中完成 add 函数

```
make_EHelper(add) {
  rtl_add(&t2, &id_dest->val, &id_src->val);
  operand_write(id_dest, &t2);
  rtl_update_ZFSF(&t2, id_dest->width);
  rtl_sltu(&t0,&t2,&id_dest->val);
  rtl_set_CF(&t0);
  rtl_xor(&t0, &id_dest->val, &id_src->val);
```

```
  rtl_not(&t0);
  rtl_xor(&t1, &id_dest->val, &t2);
  rtl_and(&t0, &t0, &t1);
  rtl_msb(&t0, &t0, id_dest->width);
  rtl_set_OF(&t0);
  print_asm_template2(add);

  print_asm_template2(add);
}
```

一下就运行成功了



去logic.c里面实现or函数，并且在gp1里填表，查表得填010位

```
make_EHelper(or) {
  rtl_or(&t0, &id_dest->val, &id_src->val);
  operand_write(id_dest, &t0);
  rtl_update_ZFSF(&t0, id_dest->width);
  rtl_set_CF(&tzero);
  rtl_set_OF(&tzero);
  print_asm_template2(or);

  print_asm_template2(or);
}
```

(2)add-longlong.c

用命令`make ARCH=x86-nemu ALL=add-longlong run`来测试`add-longlong.c`

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  100043:   89 e5                                   movl %esp,%ebp
  100045:   83 ec 08                                subl $0x8,%esp
  100048:   83 7d 08 00                             cmpl $0x0,0x8(%ebp)
  10004c:   75 0d                                   jne 10005b
  10005b:   90                                      nop
  10005c:   c9                                      leave
  10005d:   c3                                      ret
  1000f8:   83 c4 10                                addl $0x10,%esp
  1000fb:   b8 00 00 00 00                          movl $0x0,%eax
  100100:   8b 4d fc                                movl -0x4(%ebp),%ecx
  100103:   c9                                      leave
  100104:   8d 61 fc                                leal -0x4(%ecx),%esp
  100107:   c3                                      ret
  10002e:   89 45 f4                                movl %eax,-0xc(%ebp)
  100031:   83 ec 0c                                subl $0xc,%esp
  100034:   ff 75 f4                                pushl -0xc(%ebp)
  100037:   e8 d9 ff ff ff                          call 100015
  100015:   55                                      pushl %ebp
  100016:   89 e5                                   movl %esp,%ebp
  100018:   8b 45 08                                movl 0x8(%ebp),%eax
nemu: HIT GOOD TRAP at eip = 0x0010001b

  10001b:   d6                                      nemu trap (eax = 0)
(nemu)
```

直接成功

(3)bit.c

这里开始反汇编恢复正常了

6a(push)

译码函数 push_SI ， 执行函数 push ， 宽度1， push 已实现

填表

```
/* 0x68 */  EMPTY, EMPTY, IDEXW(push_SI, push, 1), EMPTY,
```

运行成功截图

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  100014:   c3                                      ret
  100029:   e8 c3 00 00 00                          call 1000f1
  1000f1:   8d 4c 24 04                             leal 0x4(%esp),%ecx
  1000f5:   83 e4 f0                                andl $0xfffffff0,%esp
  1000f8:   ff 71 fc                                pushl -0x4(%ecx)
  1000fb:   55                                      pushl %ebp
  1000fc:   89 e5                                   movl %esp,%ebp
  1000fe:   51                                      pushl %ecx
  1000ff:   83 ec 14                                subl $0x14,%esp
  100102:   c6 45 f6 aa                             movb $0xaa,-0xa(%ebp)
  100106:   6a 00                                   pushb $0x0
  100108:   8d 45 f6                                leal -0xa(%ebp),%eax
  10010b:   50                                      pushl %eax
  10010c:   e8 4d ff ff ff                          call 10005e
  10005e:   55                                      pushl %ebp
  10005f:   89 e5                                   movl %esp,%ebp
  100061:   83 ec 10                                subl $0x10,%esp
  100064:   8b 45 0c                                movl 0xc(%ebp),%eax
invalid opcode(eip = 0x00100067): c1 f8 03 89 45 fc 83 65 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x00100067 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x00100067) in the disassembling result to distinguish which case it is.
```

c1(sar)

c1为对应的表项已填写好，在 build 目录下查看反汇编文件`bit-x86-nemu.txt`可知需要执行 sar 指令

```
/* 0xc0 */ IDEXW(gp2_Ib2E, gp2, 1), IDEX(gp2_Ib2E, gp2), EMPTY, EX(ret),
```

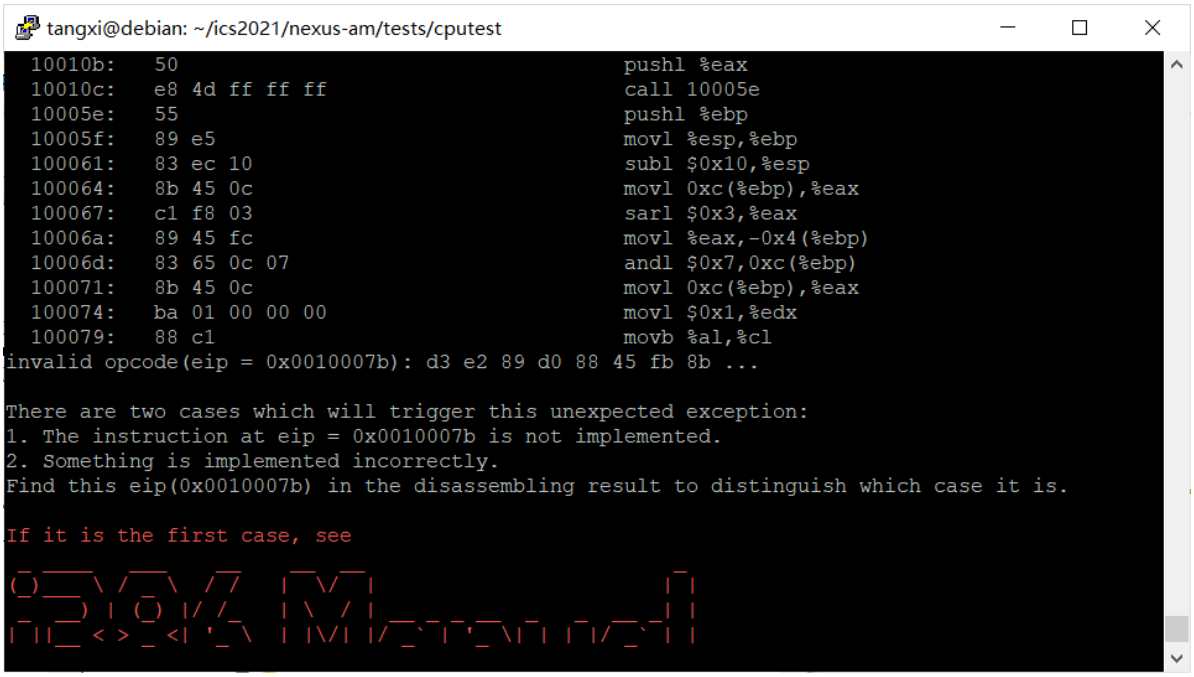
查阅i386手册附录A可知需填写第 111 项

```
make_group(gp2,
  EMPTY, EMPTY, EMPTY, EMPTY,
  EMPTY, EMPTY, EMPTY, EX(sar))
```

在 `logic.c` 文件中完成 sar 函数

```
make_EHelper(sar) {
  rtl_sar(&id_dest->val, &id_dest->val, &id_src->val);
  operand_write(id_dest, &id_dest->val);
  rtl_update_ZFSF(&id_dest->val,id_dest->width);
  // unnecessary to update CF and OF in NEMU

  print_asm_template2(sar);
}
```

运行成功截图



d3(shl)

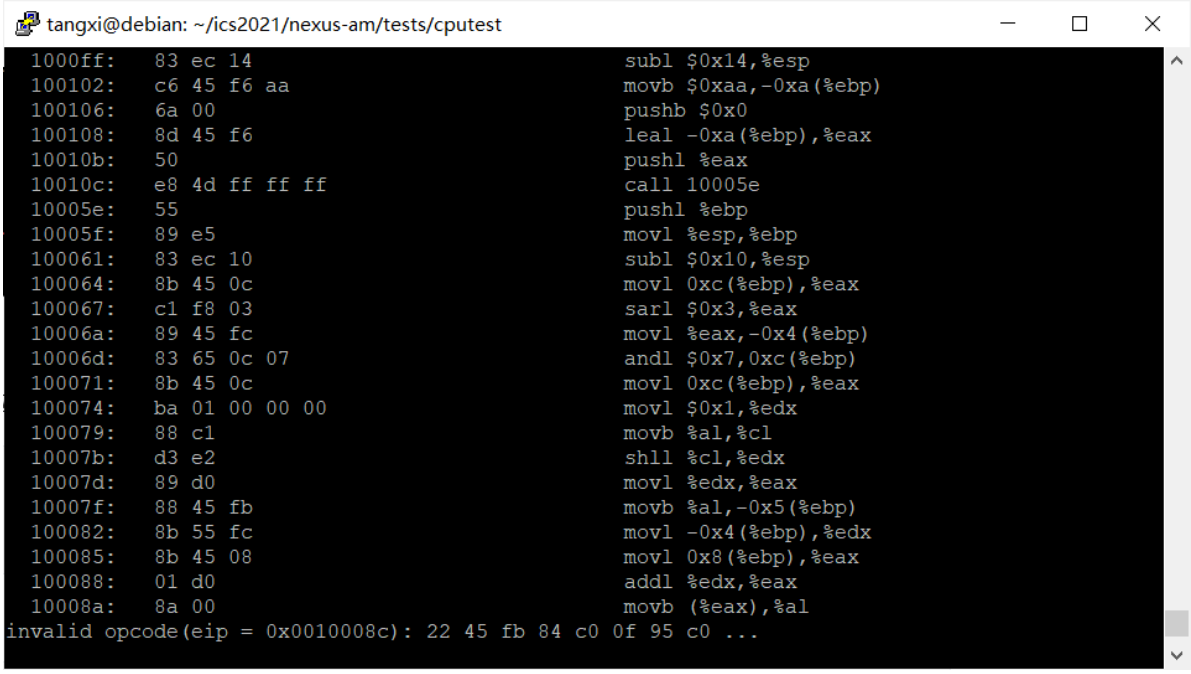
同样表已经填好了，查看手册得知要填gp2的101位

```
make_group(gp2,
    EMPTY, EMPTY, EMPTY, EMPTY,
    EX(shl), EMPTY, EMPTY, EX(sar))
```

在 `logic.c` 文件中完成 shl 函数

```
make_EHelper(shl) {
  rtl_shl(&id_dest->val, &id_dest->val, &id_src->val);
  operand_write(id_dest, &id_dest->val);
  rtl_update_ZFSF(&id_dest->val, id_dest->width);
  // unnecessary to update CF and OF in NEMU

  print_asm_template2(shl);
}
```

运行成功截图

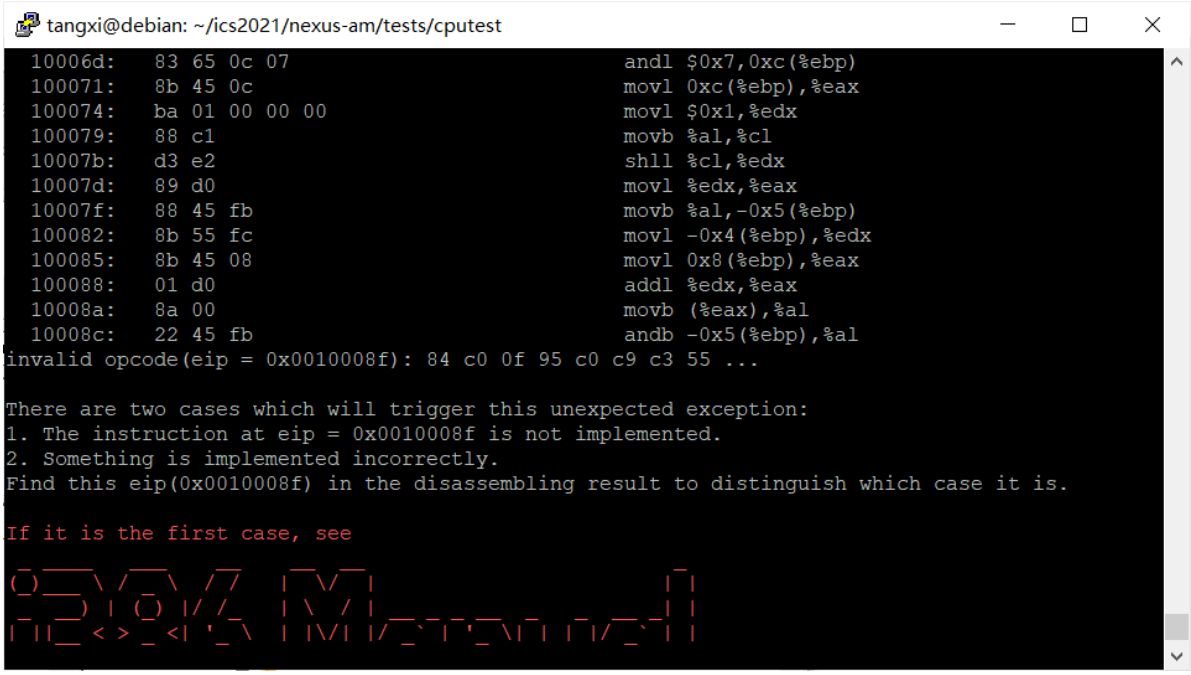


22(and)

译码函数 E2G ， 执行函数 and ， 宽度为1， and 已实现，填表

```
  /* 0x20 */    EMPTY, EMPTY, IDEXW(E2G, and, 1), EMPTY,
```

运行成功截图

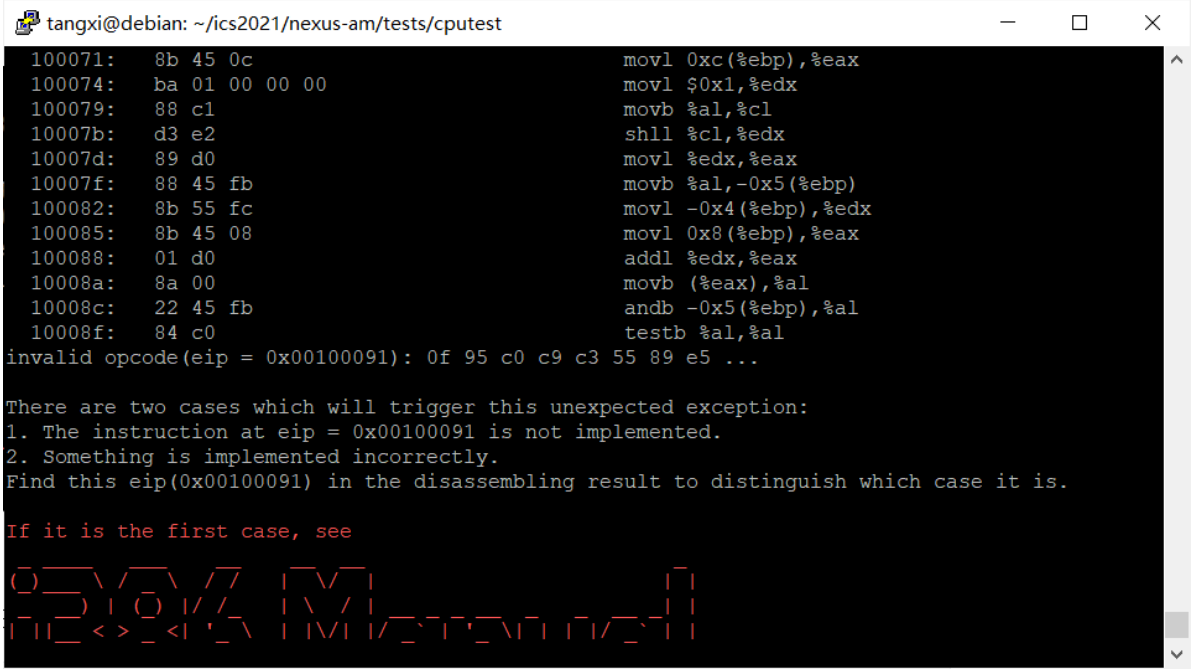


84(test)

译码函数 G2E ， 执行函数 test ， 宽度为1， test在之前就写好了，只要填表即可

```
/* 0x84 */  IDEXW(G2E, test, 1), IDEX(G2E, test), EMPTY, EMPTY,
```

运行成功截图

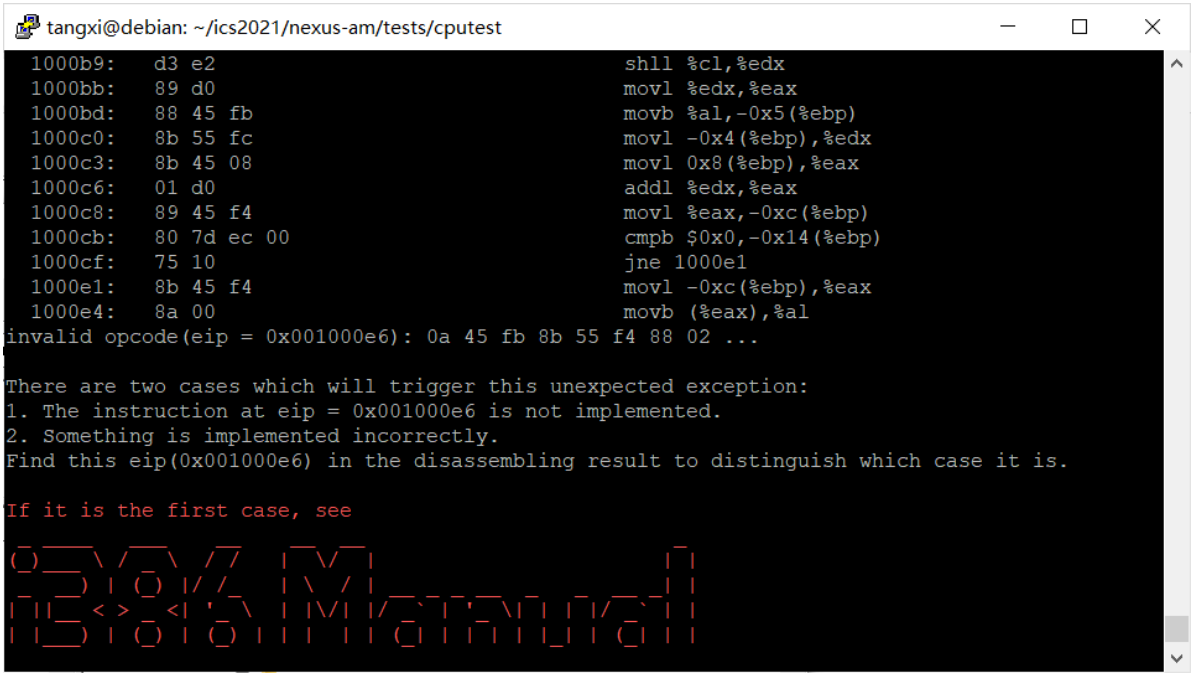


0f 95(set)

0f是2字节的表格，译码函数 E ， 执行函数 setcc ， 宽度为1， 写0f 94时 setcc 已实现，继续填表

```
/* 0x94 */  IDEXW(E, setcc, 1), IDEXW(E, setcc, 1), EMPTY, EMPTY,
```

运行成功截图

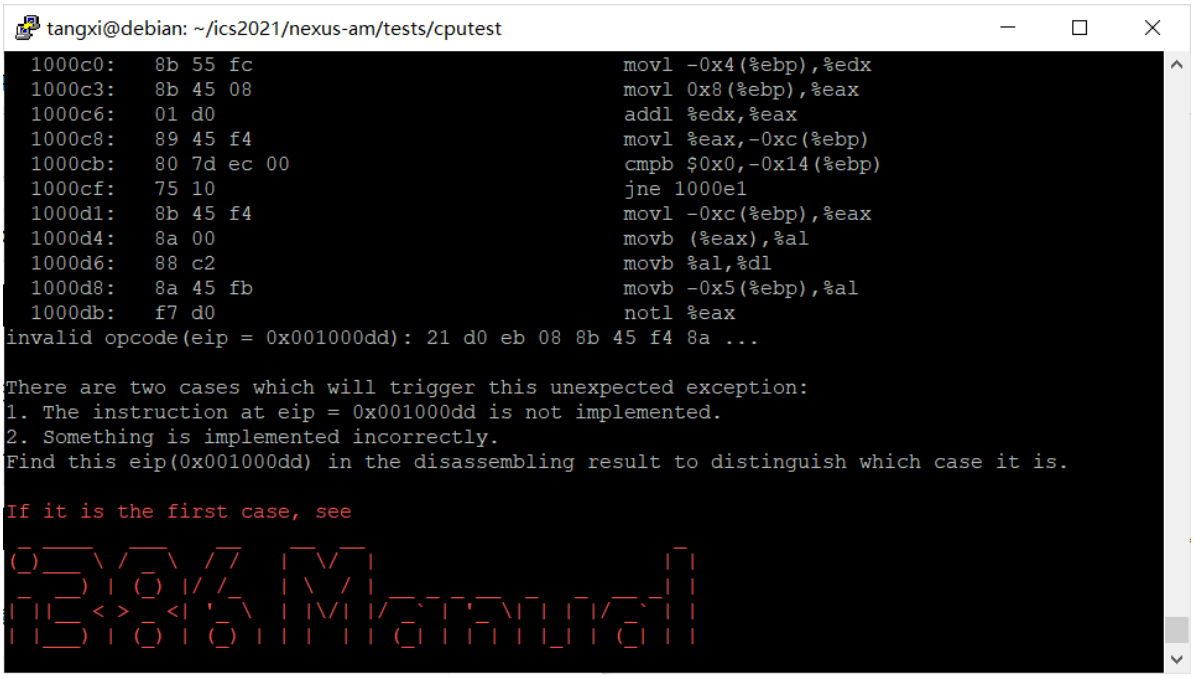


0a(or)

译码函数 G2E ， 执行函数 or ， 宽度为1， or 已实现，填表

```
/* 0x08 */    EMPTY, IDEX(G2E, or), IDEXW(E2G,or,1), EMPTY,
```

运行成功截图

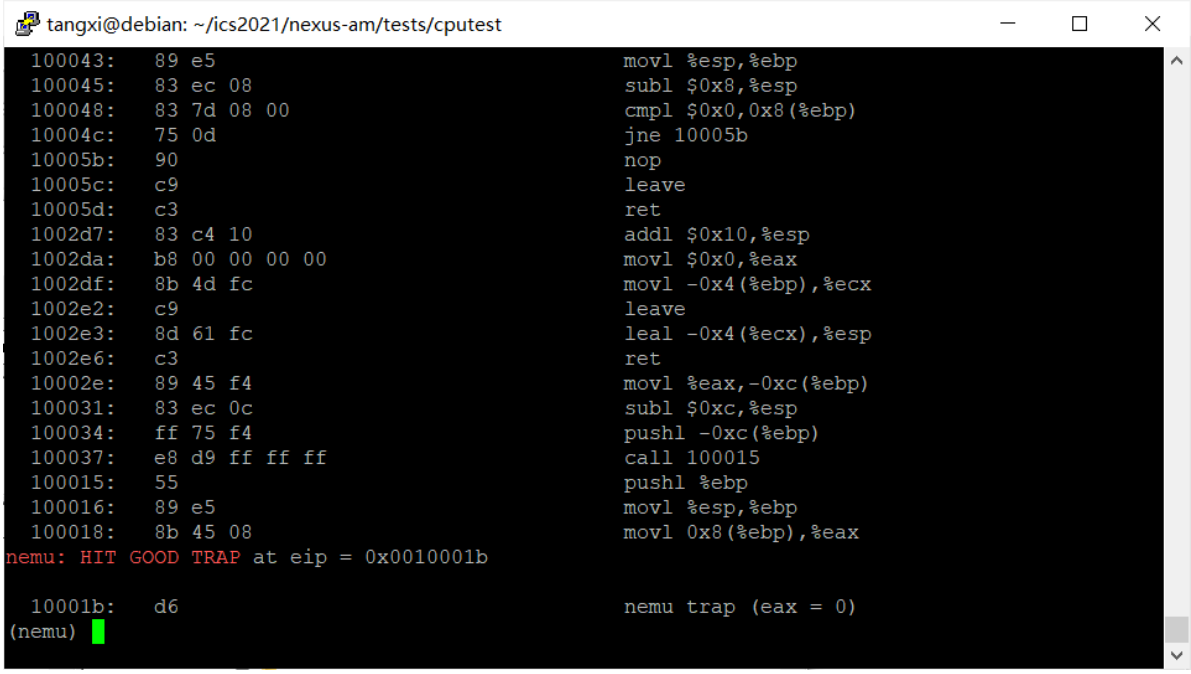


21(and)

译码函数 G2E ， 执行函数 and ， and 已实现，填表

```
/* 0x20 */ EMPTY, IDEX(G2E, and), IDEXW(E2G, and, 1), EMPTY,
```

运行成功截图



(4)bubble-sort.c

用命令`make ARCH=x86-nemu ALL=bubble-sort run`测试bubble-sort.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  1000eb:   83 ec 14                          subl $0x14,%esp
  1000ee:   e8 6b ff ff ff                    call 10005e
  10005e:   55                                pushl %ebp
  10005f:   89 e5                             movl %esp,%ebp
  100061:   83 ec 10                          subl $0x10,%esp
  100064:   c7 45 f8 00 00 00 00              movl $0x0,-0x8(%ebp)
  10006b:   eb 67                             jmp 1000d4
  1000d4:   83 7d f8 13                       cmpl $0x13,-0x8(%ebp)
  1000d8:   7e 93                             jle 10006d
  10006d:   c7 45 fc 00 00 00 00              movl $0x0,-0x4(%ebp)
  100074:   eb 4e                             jmp 1000c4
  1000c4:   b8 13 00 00 00                    movl $0x13,%eax
invalid opcode(eip = 0x001000c9): 2b 45 f8 39 45 fc 7c a5 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x001000c9 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x001000c9) in the disassembling result to distinguish which case it is.

If it is the first case, see
```

2b(sub)

译码函数 E2G ， 执行函数 sub ， sub 已实现，填表

```
/* 0x28 */ EMPTY, EMPTY, EMPTY, IDEX(E2G, sub),
```

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  1000d4:   83 7d f8 13                       cmpl $0x13,-0x8(%ebp)
  1000d8:   7e 93                             jle 10006d
  10006d:   c7 45 fc 00 00 00 00              movl $0x0,-0x4(%ebp)
  100074:   eb 4e                             jmp 1000c4
  1000c4:   b8 13 00 00 00                    movl $0x13,%eax
  1000c9:   2b 45 f8                          subl -0x8(%ebp),%eax
  1000cc:   39 45 fc                          cmpl %eax,-0x4(%ebp)
  1000cf:   7c a5                             jl 100076
  100076:   8b 45 fc                          movl -0x4(%ebp),%eax
  100079:   8b 14 85 c0 01 10 00              movl 0x1001c0(,%eax,4),%edx
  100080:   8b 45 fc                          movl -0x4(%ebp),%eax
invalid opcode(eip = 0x00100083): 40 8b 04 85 c0 01 10 00 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x00100083 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x00100083) in the disassembling result to distinguish which case it is.

If it is the first case, see
```

40(inc)

译码函数 r ， 执行函数 inc ， inc 已实现，填表

```
/* 0x40 */ IDEX(r, inc), EMPTY, EMPTY, EMPTY,
```

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  100043:   89 e5                                   movl %esp,%ebp
  100045:   83 ec 08                                subl $0x8,%esp
  100048:   83 7d 08 00                             cmpl $0x0,0x8(%ebp)
  10004c:   75 0d                                   jne 10005b
  10005b:   90                                      nop
  10005c:   c9                                      leave
  10005d:   c3                                      ret
  100183:   83 c4 10                                addl $0x10,%esp
  100186:   b8 00 00 00 00                          movl $0x0,%eax
  10018b:   8b 4d fc                                movl -0x4(%ebp),%ecx
  10018e:   c9                                      leave
  10018f:   8d 61 fc                                leal -0x4(%ecx),%esp
  100192:   c3                                      ret
  10002e:   89 45 f4                                movl %eax,-0xc(%ebp)
  100031:   83 ec 0c                                subl $0xc,%esp
  100034:   ff 75 f4                                pushl -0xc(%ebp)
  100037:   e8 d9 ff ff ff                          call 100015
  100015:   55                                      pushl %ebp
  100016:   89 e5                                   movl %esp,%ebp
  100018:   8b 45 08                                movl 0x8(%ebp),%eax
nemu: HIT GOOD TRAP at eip = 0x0010001b

  10001b:   d6                                      nemu trap (eax = 0)
(nemu)
```

(5)fact.c

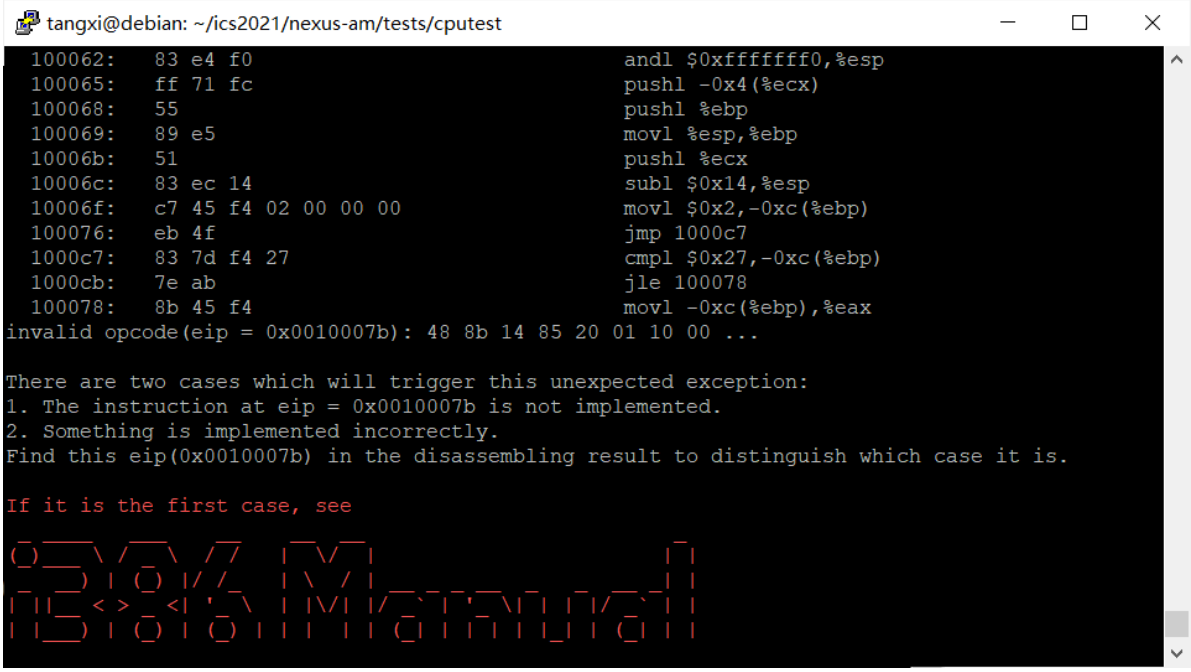
用命令`make ARCH=x86-nemu ALL=fact run`测试fact.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  100043:   89 e5                                   movl %esp,%ebp
  100045:   83 ec 08                                subl $0x8,%esp
  100048:   83 7d 08 00                             cmpl $0x0,0x8(%ebp)
  10004c:   75 0d                                   jne 10005b
  10005b:   90                                      nop
  10005c:   c9                                      leave
  10005d:   c3                                      ret
  100183:   83 c4 10                                addl $0x10,%esp
  100186:   b8 00 00 00 00                          movl $0x0,%eax
  10018b:   8b 4d fc                                movl -0x4(%ebp),%ecx
  10018e:   c9                                      leave
  10018f:   8d 61 fc                                leal -0x4(%ecx),%esp
  100192:   c3                                      ret
  10002e:   89 45 f4                                movl %eax,-0xc(%ebp)
  100031:   83 ec 0c                                subl $0xc,%esp
  100034:   ff 75 f4                                pushl -0xc(%ebp)
  100037:   e8 d9 ff ff ff                          call 100015
  100015:   55                                      pushl %ebp
  100016:   89 e5                                   movl %esp,%ebp
  100018:   8b 45 08                                movl 0x8(%ebp),%eax
nemu: HIT GOOD TRAP at eip = 0x0010001b

  10001b:   d6                                      nemu trap (eax = 0)
(nemu)
```

直接运行成功

(6)fib.c

用命令`make ARCH=x86-nemu ALL=fib run`测试fib.c



48(dec)

查表得知，译码函数 r ， 执行函数 dec ， 将48-4f填完

```
/* 0x48 */ IDEX(r, dec), IDEX(r, dec), IDEX(r, dec), IDEX(r, dec),
/* 0x4c */ IDEX(r, dec), IDEX(r, dec), IDEX(r, dec), IDEX(r, dec),
```

在 `arith.c` 文件中完成 dec 函数

```
make_EHelper(dec) {
  rtl_subi(&t2, &id_dest->val, 1);
  operand_write(id_dest, &t2);
  rtl_update_ZFSF(&t2, id_dest->width);
  rtl_xor(&t0, &id_dest->val, &id_src->val);
  rtl_xor(&t1, &id_dest->val, &t2);
  rtl_and(&t0, &t0, &t1);
  rtl_msb(&t0, &t0, id_dest->width);
  rtl_set_OF(&t0);

  print_asm_template1(dec);
}
```

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  100043:   89 e5                           movl %esp,%ebp
  100045:   83 ec 08                        subl $0x8,%esp
  100048:   83 7d 08 00                     cmpl $0x0,0x8(%ebp)
  10004c:   75 0d                           jne 10005b
  10005b:   90                              nop
  10005c:   c9                              leave
  10005d:   c3                              ret
  1000e0:   83 c4 10                        addl $0x10,%esp
  1000e3:   b8 00 00 00 00                  movl $0x0,%eax
  1000e8:   8b 4d fc                        movl -0x4(%ebp),%ecx
  1000eb:   c9                              leave
  1000ec:   8d 61 fc                        leal -0x4(%ecx),%esp
  1000ef:   c3                              ret
  10002e:   89 45 f4                        movl %eax,-0xc(%ebp)
  100031:   83 ec 0c                        subl $0xc,%esp
  100034:   ff 75 f4                        pushl -0xc(%ebp)
  100037:   e8 d9 ff ff ff                  call 100015
  100015:   55                              pushl %ebp
  100016:   89 e5                           movl %esp,%ebp
  100018:   8b 45 08                        movl 0x8(%ebp),%eax
nemu: HIT GOOD TRAP at eip = 0x0010001b

  10001b:   d6                              nemu trap (eax = 0)
(nemu)
```

(7)goldbach.c

用命令`make ARCH=x86-nemu ALL=goldbach run`测试goldbach.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  10004c:   75 0d                           jne 10005b
  10005b:   90                              nop
  10005c:   c9                              leave
  10005d:   c3                              ret
  100121:   83 c4 10                        addl $0x10,%esp
  100124:   83 45 f4 02                     addl $0x2,-0xc(%ebp)
  100128:   83 7d f4 1e                     cmpl $0x1e,-0xc(%ebp)
  10012c:   7e d6                           jle 100104
  10012e:   b8 00 00 00 00                  movl $0x0,%eax
  100133:   8b 4d fc                        movl -0x4(%ebp),%ecx
  100136:   c9                              leave
  100137:   8d 61 fc                        leal -0x4(%ecx),%esp
  10013a:   c3                              ret
  10002e:   89 45 f4                        movl %eax,-0xc(%ebp)
  100031:   83 ec 0c                        subl $0xc,%esp
  100034:   ff 75 f4                        pushl -0xc(%ebp)
  100037:   e8 d9 ff ff ff                  call 100015
  100015:   55                              pushl %ebp
  100016:   89 e5                           movl %esp,%ebp
  100018:   8b 45 08                        movl 0x8(%ebp),%eax
nemu: HIT GOOD TRAP at eip = 0x0010001b

  10001b:   d6                              nemu trap (eax = 0)
(nemu)
```

直接运行成功

(8)hello-str.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  100013:   5d                                  popl %ebp
  100014:   c3                                  ret
  100038:   e8 30 00 00 00                      call 10006d
  10006d:   8d 4c 24 04                         leal 0x4(%esp),%ecx
  100071:   83 e4 f0                            andl $0xfffffff0,%esp
  100074:   ff 71 fc                            pushl -0x4(%ecx)
  100077:   55                                  pushl %ebp
  100078:   89 e5                               movl %esp,%ebp
  10007a:   51                                  pushl %ecx
  10007b:   83 ec 04                            subl $0x4,%esp
  10007e:   83 ec 04                            subl $0x4,%esp
invalid opcode(eip = 0x00100081): 68 70 08 10 00 68 7e 08 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x00100081 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x00100081) in the disassembling result to distinguish which case it is.

If it is the first case, see
```

68(push)

译码函数 push_SI ， 执行函数 push ， push 已实现，填表

```
/* 0x68 */ IDEX(push_SI, push), EMPTY, IDEXW(push_SI, push, 1), EMPTY,
```

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  10080a:   ff 75 0c                            pushl 0xc(%ebp)
  10080d:   ff 75 08                            pushl 0x8(%ebp)
  100810:   e8 b7 fd ff ff                      call 1005cc
  1005cc:   55                                  pushl %ebp
  1005cd:   89 e5                               movl %esp,%ebp
  1005cf:   57                                  pushl %edi
  1005d0:   56                                  pushl %esi
  1005d1:   53                                  pushl %ebx
  1005d2:   83 ec 1c                            subl $0x1c,%esp
  1005d5:   8b 75 0c                            movl 0xc(%ebp),%esi
  1005d8:   31 ff                               xorl %edi,%edi
invalid opcode(eip = 0x001005da): 0f be 06 84 c0 74 1d 3c ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x001005da is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x001005da) in the disassembling result to distinguish which case it is.

If it is the first case, see
```

0f be(movsx)

查看反汇编可知需实现 movsx 指令

译码函数 mov_E2G ， 执行函数 movsx ， 宽度1，发现 movsx 已实现，填表

```
/* 0xbc */ EMPTY, EMPTY, IDEXW(mov_E2G, movsx, 1), EMPTY,
```

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  1005cc:   55                                  pushl %ebp
  1005cd:   89 e5                               movl %esp,%ebp
  1005cf:   57                                  pushl %edi
  1005d0:   56                                  pushl %esi
  1005d1:   53                                  pushl %ebx
  1005d2:   83 ec 1c                            subl $0x1c,%esp
  1005d5:   8b 75 0c                            movl 0xc(%ebp),%esi
  1005d8:   31 ff                               xorl %edi,%edi
  1005da:   0f be 06                            movsxl (%esi),%al
  1005dd:   84 c0                               testb %al,%al
  1005df:   74 1d                               je 1005fe
  1005e1:   3c 25                               cmpb $0x25,%al
  1005e3:   74 2f                               je 100614
  100614:   0f be 46 01                         movsxl 0x1(%esi),%al
  100618:   bb ff ff ff ff                      movl $0xffffffff,%ebx
  10061d:   c6 45 e4 20                         movb $0x20,-0x1c(%ebp)
  100621:   8d 4e 01                            leal 0x1(%esi),%ecx
  100624:   8d 50 e0                            leal -0x20(%eax),%edx
  100627:   80 fa 58                            cmpb $0x58,%dl
  10062a:   77 72                               jnbe 10069e
  10062c:   0f b6 d2                            movzxl %dl,%dl
  10062f:   ff 24 95 a8 08 10 00                jmp 10069e
QEMU eip:0x00100714 NEMU eip:0x0010069e
(nemu)
```

ff(jmp)

ff已经写好了，

```
/* 0xfc */ EMPTY, EMPTY, IDEXW(E, gp4, 1), IDEX(E, gp5),
```

发现diff报错，是jmp没写好，查看反汇编得知ff (jmp)无译码函数，执行函数 jmp_rm ， jmp_rm 已实现，在 all-instr.h 中声明 make_EHelper(jmp_rm) ，查阅i386手册附录A可知需填写第 100 项

```
make_group(gp5,
    EX(inc), EMPTY, EMPTY, EX(call),
    EX(jmp_rm), EX(jmp), EX(push), EMPTY)
```

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  100590:   55                                  pushl %ebp
  100591:   89 e5                               movl %esp,%ebp
  100593:   56                                  pushl %esi
  100594:   53                                  pushl %ebx
  100595:   8b 5d 08                            movl 0x8(%ebp),%ebx
  100598:   8b 75 0c                            movl 0xc(%ebp),%esi
  10059b:   8b 06                               movl (%esi),%eax
  10059d:   eb 0d                               jmp 1005ac
  1005ac:   0f be 13                            movsxl (%ebx),%dl
  1005af:   84 d2                               testb %dl,%dl
  1005b1:   75 ed                               jne 1005a0
invalid opcode(eip = 0x001005a0): 43 85 c0 74 17 88 10 8b ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x001005a0 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x001005a0) in the disassembling result to distinguish which case it is.

If it is the first case, see
```

43(inc)

译码函数 r ， 执行函数 inc ， inc 已实现，填完40-47

```
/* 0x40 */    IDEX(r, inc), IDEX(r, inc), IDEX(r, inc), IDEX(r, inc),
/* 0x44 */    IDEX(r, inc), IDEX(r, inc), IDEX(r, inc), IDEX(r, inc),
```

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  100857:   84 c0                                   testb %al,%al
  100859:   74 0d                                   je 100868
  10085b:   89 f3                                   movl %esi,%ebx
  10085d:   38 d0                                   cmpb %dl,%al
  10085f:   74 eb                                   je 10084c
  10084c:   41                                      incl %ecx
  10084d:   8d 73 01                                leal 0x1(%ebx),%esi
  100850:   0f be 01                                movsxl (%ecx),%al
  100853:   0f be 53 01                             movsxl 0x1(%ebx),%dl
  100857:   84 c0                                   testb %al,%al
  100859:   74 0d                                   je 100868
  100868:   31 c0                                   xorl %eax,%eax
invalid opcode(eip = 0x0010086a): 29 d0 5b 5e 5d c3 48 65 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x0010086a is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x0010086a) in the disassembling result to distinguish which case it is.

If it is the first case, see
```

29(sub)

译码函数 G2E ， 执行函数 sub ， sub 已实现，填表

```
/* 0x28 */ EMPTY, IDEX(G2E, sub), EMPTY, IDEX(E2G, sub),
```

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  100052:   89 e5                                   movl %esp,%ebp
  100054:   83 ec 08                                subl $0x8,%esp
  100057:   83 7d 08 00                             cmpl $0x0,0x8(%ebp)
  10005b:   75 0d                                   jne 10006a
  10006a:   90                                      nop
  10006b:   c9                                      leave
  10006c:   c3                                      ret
  100146:   83 c4 10                                addl $0x10,%esp
  100149:   b8 00 00 00 00                          movl $0x0,%eax
  10014e:   8b 4d fc                                movl -0x4(%ebp),%ecx
  100151:   c9                                      leave
  100152:   8d 61 fc                                leal -0x4(%ecx),%esp
  100155:   c3                                      ret
  10003d:   89 45 f4                                movl %eax,-0xc(%ebp)
  100040:   83 ec 0c                                subl $0xc,%esp
  100043:   ff 75 f4                                pushl -0xc(%ebp)
  100046:   e8 d9 ff ff ff                          call 100024
  100024:   55                                      pushl %ebp
  100025:   89 e5                                   movl %esp,%ebp
  100027:   8b 45 08                                movl 0x8(%ebp),%eax
nemu: HIT GOOD TRAP at eip = 0x0010002a

  10002a:   d6                                      nemu trap (eax = 0)
(nemu)
```

(9)if-else.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  100043:   89 e5                                   movl %esp,%ebp
  100045:   83 ec 08                                subl $0x8,%esp
  100048:   83 7d 08 00                             cmpl $0x0,0x8(%ebp)
  10004c:   75 0d                                   jne 10005b
  10005b:   90                                      nop
  10005c:   c9                                      leave
  10005d:   c3                                      ret
  10012a:   83 c4 10                                addl $0x10,%esp
  10012d:   b8 00 00 00 00                          movl $0x0,%eax
  100132:   8b 4d fc                                movl -0x4(%ebp),%ecx
  100135:   c9                                      leave
  100136:   8d 61 fc                                leal -0x4(%ecx),%esp
  100139:   c3                                      ret
  10002e:   89 45 f4                                movl %eax,-0xc(%ebp)
  100031:   83 ec 0c                                subl $0xc,%esp
  100034:   ff 75 f4                                pushl -0xc(%ebp)
  100037:   e8 d9 ff ff ff                          call 100015
  100015:   55                                      pushl %ebp
  100016:   89 e5                                   movl %esp,%ebp
  100018:   8b 45 08                                movl 0x8(%ebp),%eax
nemu: HIT GOOD TRAP at eip = 0x0010001b

  10001b:   d6                                      nemu trap (eax = 0)
(nemu)
```

直接运行成功

(10)leap-year.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  10009f:   83 e4 f0                                andl $0xfffffff0,%esp
  1000a2:   ff 71 fc                                pushl -0x4(%ecx)
  1000a5:   55                                      pushl %ebp
  1000a6:   89 e5                                   movl %esp,%ebp
  1000a8:   51                                      pushl %ecx
  1000a9:   83 ec 14                                subl $0x14,%esp
  1000ac:   c7 45 f4 00 00 00 00                    movl $0x0,-0xc(%ebp)
  1000b3:   eb 34                                   jmp 1000e9
  1000e9:   83 7d f4 7c                             cmpl $0x7c,-0xc(%ebp)
  1000ed:   7e c6                                   jle 1000b5
  1000b5:   8b 45 f4                                movl -0xc(%ebp),%eax
invalid opcode(eip = 0x001000b8): 05 62 07 00 00 50 e8 9b ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x001000b8 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x001000b8) in the disassembling result to distinguish which case it is.

If it is the first case, see
```

05(add)

译码函数 I2a ，执行函数 add ， add 已实现，填表

```
/* 0x04 */ EMPTY, IDEX(I2a, add), EMPTY, EMPTY,
```

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  100085:   f7 f9                                   idivl %ecx
  100087:   89 d0                                   movl %edx,%eax
  100089:   85 c0                                   testl %eax,%eax
  10008b:   75 07                                   jne 100094
  100094:   b8 00 00 00 00                          movl $0x0,%eax
  100099:   5d                                      popl %ebp
  10009a:   c3                                      ret
  1000c3:   83 c4 04                                addl $0x4,%esp
  1000c6:   89 c2                                   movl %eax,%edx
  1000c8:   8b 45 f4                                movl -0xc(%ebp),%eax
  1000cb:   8b 04 85 20 01 10 00                    movl 0x100120(,%eax,4),%eax
  1000d2:   39 c2                                   cmpl %eax,%edx
  1000d4:   0f 94 c0                                sete %al
  1000d7:   0f b6 c0                                movzxl %al,%al
  1000da:   83 ec 0c                                subl $0xc,%esp
  1000dd:   50                                      pushl %eax
  1000de:   e8 5f ff ff ff                          call 100042
  100042:   55                                      pushl %ebp
  100043:   89 e5                                   movl %esp,%ebp
  100045:   83 ec 08                                subl $0x8,%esp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

(11)load-store.c

0f bf(movsx)

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  100065:   ff 71 fc                                pushl -0x4(%ecx)
  100068:   55                                      pushl %ebp
  100069:   89 e5                                   movl %esp,%ebp
  10006b:   51                                      pushl %ecx
  10006c:   83 ec 14                                subl $0x14,%esp
  10006f:   c7 45 f4 00 00 00 00                    movl $0x0,-0xc(%ebp)
  100076:   eb 2f                                   jmp 1000a7
  1000a7:   83 7d f4 07                             cmpl $0x7,-0xc(%ebp)
  1000ab:   76 cb                                   jbe 100078
  100078:   8b 45 f4                                movl -0xc(%ebp),%eax
  10007b:   66 8b 84 00 c0 01 10 00                 movw 0x1001c0(%eax,%eax,1),%ax
invalid opcode(eip = 0x00100083): 0f bf d0 8b 45 f4 8b 04 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x00100083 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x00100083) in the disassembling result to distinguish which case it is.

If it is the first case, see
```

译码函数 mov_E2G ， 执行函数 movsx ， 宽度2， movsx 已实现，填表

```
/* 0xbc */ EMPTY, EMPTY, IDEXW(mov_E2G, movsx, 1), IDEXW(mov_E2G, movsx,
2),
```

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  10005d:   c3                                  ret
  1000a1:   83 c4 10                            addl $0x10,%esp
  1000a4:   ff 45 f4                            incl -0xc(%ebp)
  1000a7:   83 7d f4 07                         cmpl $0x7,-0xc(%ebp)
  1000ab:   76 cb                               jbe 100078
  1000ad:   c7 45 f4 00 00 00 00                movl $0x0,-0xc(%ebp)
  1000b4:   eb 2f                               jmp 1000e5
  1000e5:   83 7d f4 07                         cmpl $0x7,-0xc(%ebp)
  1000e9:   76 cb                               jbe 1000b6
  1000b6:   8b 45 f4                            movl -0xc(%ebp),%eax
  1000b9:   66 8b 84 00 c0 01 10 00             movw 0x1001c0(%eax,%eax,1),%ax
invalid opcode(eip = 0x001000c1): 0f b7 d0 8b 45 f4 8b 04 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x001000c1 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x001000c1) in the disassembling result to distinguish which case it is.

If it is the first case, see
```

0f b7(movzx)

译码函数 mov_E2G ，执行函数 movzx ，宽度2， movzx 已实现，填表

```
/* 0xb4 */ EMPTY, EMPTY, IDEXW(mov_E2G, movzx, 1), IDEXW(mov_E2G, movzx,
2),
```

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  10004c:   75 0d                               jne 10005b
  10005b:   90                                  nop
  10005c:   c9                                  leave
  10005d:   c3                                  ret
  10017d:   83 c4 10                            addl $0x10,%esp
  100180:   ff 45 f4                            incl -0xc(%ebp)
  100183:   83 7d f4 07                         cmpl $0x7,-0xc(%ebp)
  100187:   76 ad                               jbe 100136
  100189:   b8 00 00 00 00                      movl $0x0,%eax
  10018e:   8b 4d fc                            movl -0x4(%ebp),%ecx
  100191:   c9                                  leave
  100192:   8d 61 fc                            leal -0x4(%ecx),%esp
  100195:   c3                                  ret
  10002e:   89 45 f4                            movl %eax,-0xc(%ebp)
  100031:   83 ec 0c                            subl $0xc,%esp
  100034:   ff 75 f4                            pushl -0xc(%ebp)
  100037:   e8 d9 ff ff ff                      call 100015
  100015:   55                                  pushl %ebp
  100016:   89 e5                               movl %esp,%ebp
  100018:   8b 45 08                            movl 0x8(%ebp),%eax
nemu: HIT GOOD TRAP at eip = 0x0010001b

  10001b:   d6                                  nemu trap (eax = 0)
(nemu)
```

(12)matrix-mul.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  1000d1:   8b 55 ec                          movl -0x14(%ebp),%edx
  1000d4:   01 d0                             addl %edx,%eax
  1000d6:   8b 1c 85 00 02 10 00              movl 0x100200(,%eax,4),%ebx
  1000dd:   8b 55 ec                          movl -0x14(%ebp),%edx
  1000e0:   89 d0                             movl %edx,%eax
  1000e2:   c1 e0 02                          shll $0x2,%eax
  1000e5:   01 d0                             addl %edx,%eax
  1000e7:   01 c0                             addl %eax,%eax
  1000e9:   8b 55 f0                          movl -0x10(%ebp),%edx
  1000ec:   01 d0                             addl %edx,%eax
  1000ee:   8b 04 85 a0 03 10 00              movl 0x1003a0(,%eax,4),%eax
invalid opcode(eip = 0x001000f5): 0f af c3 01 c1 8b 55 f4 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x001000f5 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x001000f5) in the disassembling result to distinguish which case it is.

If it is the first case, see
```

0f af(imul)

0f是2字节表，译码函数 E2G ，所以执行函数 imul2 ， imul2 已实现，填表

```
/* 0xac */ EMPTY, EMPTY, EMPTY, IDEX(E2G,imul2),
```

运行成功截图

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/matrix-mul-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si10
Unknown command 'si10'
(nemu) si 10
  100000:   bd 00 00 00 00                    movl $0x0,%ebp
  100005:   bc 00 7c 00 00                    movl $0x7c00,%esp
  10000a:   e8 0f 00 00 00                    call 10001e
  10001e:   55                                pushl %ebp
  10001f:   89 e5                             movl %esp,%ebp
  100021:   83 ec 18                          subl $0x18,%esp
  100024:   e8 e6 ff ff ff                    call 10000f
  10000f:   55                                pushl %ebp
  100010:   89 e5                             movl %esp,%ebp
  100012:   90                                nop
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

(13)max.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  1000be:   50                                  pushl %eax
  1000bf:   e8 9a ff ff ff                      call 10005e
  10005e:   55                                  pushl %ebp
  10005f:   89 e5                               movl %esp,%ebp
  100061:   83 ec 10                            subl $0x10,%esp
  100064:   8b 45 08                            movl 0x8(%ebp),%eax
  100067:   3b 45 0c                            cmpl 0xc(%ebp),%eax
  10006a:   7e 08                               jle 100074
  100074:   8b 45 0c                            movl 0xc(%ebp),%eax
  100077:   89 45 fc                            movl %eax,-0x4(%ebp)
  10007a:   8b 45 fc                            movl -0x4(%ebp),%eax
  10007d:   c9                                  leave
  10007e:   c3                                  ret
  1000c4:   83 c4 08                            addl $0x8,%esp
  1000c7:   89 c1                               movl %eax,%ecx
  1000c9:   8b 45 ec                            movl -0x14(%ebp),%eax
  1000cc:   8d 50 01                            leal 0x1(%eax),%edx
  1000cf:   89 55 ec                            movl %edx,-0x14(%ebp)
  1000d2:   8b 04 85 80 01 10 00                movl 0x100180(,%eax,4),%eax
  1000d9:   39 c1                               cmpl %eax,%ecx
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

直接通过

(14)min3.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/min3-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si9
Unknown command 'si9'
(nemu) si 9
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
  100005:   bc 00 7c 00 00                      movl $0x7c00,%esp
  10000a:   e8 0f 00 00 00                      call 10001e
  10001e:   55                                  pushl %ebp
  10001f:   89 e5                               movl %esp,%ebp
  100021:   83 ec 18                            subl $0x18,%esp
  100024:   e8 e6 ff ff ff                      call 10000f
  10000f:   55                                  pushl %ebp
  100010:   89 e5                               movl %esp,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

直接运行通过

(15) mov-c.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
make ARCH=x86-nemu ALL=mov-c run
Building mov-c [x86-nemu]
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/mov-c-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 9
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
  100005:   bc 00 7c 00 00                      movl $0x7c00,%esp
  10000a:   e8 0f 00 00 00                      call 10001e
  10001e:   55                                  pushl %ebp
  10001f:   89 e5                               movl %esp,%ebp
  100021:   83 ec 18                            subl $0x18,%esp
  100024:   e8 e6 ff ff ff                      call 10000f
  10000f:   55                                  pushl %ebp
  100010:   89 e5                               movl %esp,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

直接运行通过

(16)movsx.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
tangxi@debian:~/ics2021/nexus-am/tests/cputest$ make ARCH=x86-nemu ALL=movsx run
Building movsx [x86-nemu]
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/movsx-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 9
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
  100005:   bc 00 7c 00 00                      movl $0x7c00,%esp
  10000a:   e8 0f 00 00 00                      call 10001e
  10001e:   55                                  pushl %ebp
  10001f:   89 e5                               movl %esp,%ebp
  100021:   83 ec 18                            subl $0x18,%esp
  100024:   e8 e6 ff ff ff                      call 10000f
  10000f:   55                                  pushl %ebp
  100010:   89 e5                               movl %esp,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

直接运行通过

(17) mul-longlong.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 9
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
  100005:   bc 00 7c 00 00                      movl $0x7c00,%esp
  10000a:   e8 0f 00 00 00                      call 10001e
  10001e:   55                                  pushl %ebp
  10001f:   89 e5                               movl %esp,%ebp
  100021:   83 ec 18                            subl $0x18,%esp
  100024:   e8 e6 ff ff ff                      call 10000f
  10000f:   55                                  pushl %ebp
  100010:   89 e5                               movl %esp,%ebp
(nemu) c
invalid opcode(eip = 0x00100134): 0b 45 d4 85 c0 0f 94 c0 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x00100134 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x00100134) in the disassembling result to distinguish which case it is.

If it is the first case, see
 _ ____   ___    __    __ __                          _
(_)___ \ / _ \  / /   |  \/  |                        | |
 _  __) | (_) |/ /_   | \  / | __ _ _ __  _   _  __ _| |
```

0b(or)

译码函数 E2G， 执行函数 or， or 已实现，填表

```
/* 0x08 */ EMPTY, IDEX(G2E, or), IDEXW(E2G,or,1), IDEX(E2G, or),
```

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
+ CC src/cpu/exec/exec.c
+ LD build/nemu
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/mul-longlong-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 9
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
  100005:   bc 00 7c 00 00                      movl $0x7c00,%esp
  10000a:   e8 0f 00 00 00                      call 10001e
  10001e:   55                                  pushl %ebp
  10001f:   89 e5                               movl %esp,%ebp
  100021:   83 ec 18                            subl $0x18,%esp
  100024:   e8 e6 ff ff ff                      call 10000f
  10000f:   55                                  pushl %ebp
  100010:   89 e5                               movl %esp,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

(18)pascal.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
tangxi@debian:~/ics2021/nexus-am/tests/cputest$ make ARCH=x86-nemu ALL=pascal run
Building pascal [x86-nemu]
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/pascal-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 9
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
  100005:   bc 00 7c 00 00                      movl $0x7c00,%esp
  10000a:   e8 0f 00 00 00                      call 10001e
  10001e:   55                                  pushl %ebp
  10001f:   89 e5                               movl %esp,%ebp
  100021:   83 ec 18                            subl $0x18,%esp
  100024:   e8 e6 ff ff ff                      call 10000f
  10000f:   55                                  pushl %ebp
  100010:   89 e5                               movl %esp,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

直接运行成功

(19)prime.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
tangxi@debian:~/ics2021/nexus-am/tests/cputest$ make ARCH=x86-nemu ALL=prime run
Building prime [x86-nemu]
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/prime-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 9
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
  100005:   bc 00 7c 00 00                      movl $0x7c00,%esp
  10000a:   e8 0f 00 00 00                      call 10001e
  10001e:   55                                  pushl %ebp
  10001f:   89 e5                               movl %esp,%ebp
  100021:   83 ec 18                            subl $0x18,%esp
  100024:   e8 e6 ff ff ff                      call 10000f
  10000f:   55                                  pushl %ebp
  100010:   89 e5                               movl %esp,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

直接运行成功

(20)quick-sort.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 9
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
  100005:   bc 00 7c 00 00                      movl $0x7c00,%esp
  10000a:   e8 0f 00 00 00                      call 10001e
  10001e:   55                                  pushl %ebp
  10001f:   89 e5                               movl %esp,%ebp
  100021:   83 ec 18                            subl $0x18,%esp
  100024:   e8 e6 ff ff ff                      call 10000f
  10000f:   55                                  pushl %ebp
  100010:   89 e5                               movl %esp,%ebp
(nemu) c
invalid opcode(eip = 0x00100089): ff 4d f8 8b 45 fc 3b 45 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x00100089 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x00100089) in the disassembling result to distinguish which case it is.

If it is the first case, see
```

ff(dec)

查看反汇编可知，是gp5里的001，需实现 dec 指令，填表

```
make_group(gp5,
EX(inc), EX(dec), EMPTY, EX(call),
EX(jmp_rm), EMPTY, EX(push), EMPTY)
```

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
tangxi@debian:~/ics2021/nexus-am/tests/cputest$ make ARCH=x86-nemu ALL=quick-sort run
Building quick-sort [x86-nemu]
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/quick-sort-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 9
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
  100005:   bc 00 7c 00 00                      movl $0x7c00,%esp
  10000a:   e8 0f 00 00 00                      call 10001e
  10001e:   55                                  pushl %ebp
  10001f:   89 e5                               movl %esp,%ebp
  100021:   83 ec 18                            subl $0x18,%esp
  100024:   e8 e6 ff ff ff                      call 10000f
  10000f:   55                                  pushl %ebp
  100010:   89 e5                               movl %esp,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

(21)recursion.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
For help, type "help"
(nemu) si 9
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
  100005:   bc 00 7c 00 00                      movl $0x7c00,%esp
  10000a:   e8 0f 00 00 00                      call 10001e
  10001e:   55                                  pushl %ebp
  10001f:   89 e5                               movl %esp,%ebp
  100021:   83 ec 18                            subl $0x18,%esp
  100024:   e8 e6 ff ff ff                      call 10000f
  10000f:   55                                  pushl %ebp
  100010:   89 e5                               movl %esp,%ebp
(nemu) c
invalid opcode(eip = 0x001001f3): ff d0 83 c4 10 89 45 f4 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x001001f3 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x001001f3) in the disassembling result to distinguish which case it is.

If it is the first case, see
```

ff(call)

查看反汇编可知需实现 call_rm 指令,无译码函数，执行函数 call_rm ， 查阅i386手册附录A可知需填写第 010 项,填表

```
make_group(gp5,
EX(inc), EX(dec), EX(call_rm), EX(call),
EX(jmp_rm), EMPTY, EX(push), EMPTY)
```

在 `control.c` 中完成 call_rm 函数

```
make_EHelper(call_rm) {
  rtl_push(&decoding.seq_eip);
  decoding.is_jmp = 1;
  decoding.jmp_eip = id_dest->val;

  print_asm("call *%s", id_dest->str);
}
```

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
For help, type "help"
(nemu) si 9
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
  100005:   bc 00 7c 00 00                      movl $0x7c00,%esp
  10000a:   e8 0f 00 00 00                      call 10001e
  10001e:   55                                  pushl %ebp
  10001f:   89 e5                               movl %esp,%ebp
  100021:   83 ec 18                            subl $0x18,%esp
  100024:   e8 e6 ff ff ff                      call 10000f
  10000f:   55                                  pushl %ebp
  100010:   89 e5                               movl %esp,%ebp
(nemu) c
invalid opcode(eip = 0x00100188): c1 eb 1f 01 da d1 fa 83 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x00100188 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x00100188) in the disassembling result to distinguish which case it is.

If it is the first case, see
```

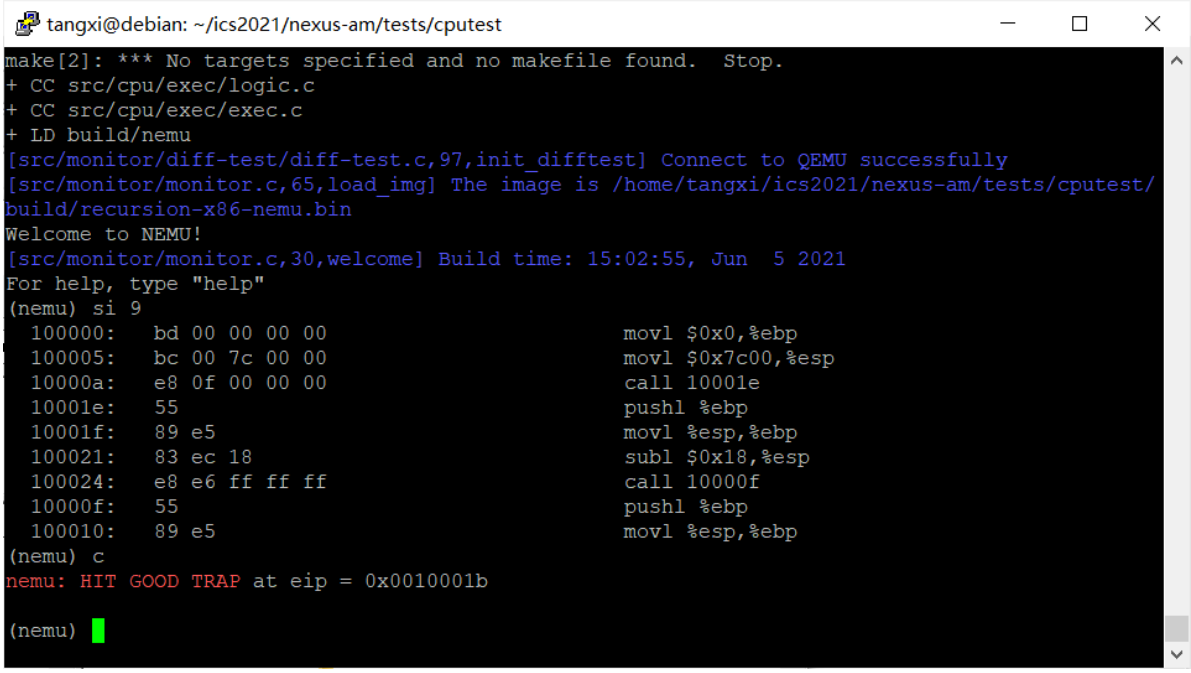
c1(shr)

查看反汇编可知需实现 shr 指令，无译码函数，执行函数 shr ，查阅i386手册附录A可知需填写第 101 项，填表

```
make_group(gp2,
EMPTY, EMPTY, EMPTY, EMPTY,
EX(shl), EX(shr), EMPTY, EX(sar))
```

在 `logic.c` 中完成 shr

```
make_EHelper(shr) {
  rtl_shr(&id_dest->val, &id_dest->val, &id_src->val);
  operand_write(id_dest, &id_dest->val);
  rtl_update_ZFSF(&id_dest->val, id_dest->width);
  // unnecessary to update CF and OF in NEMU

  print_asm_template2(shr);
}
```



(22)select-sort.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu) q
qemu-system-i386: terminating on signal 15 from pid 15608 (<unknown process>)
 recursion
tangxi@debian:~/ics2021/nexus-am/tests/cputest$ make ARCH=x86-nemu ALL=select-sort run
Building select-sort [x86-nemu]
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/select-sort-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) ai 1
Unknown command 'ai'
(nemu) si 1
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

直接运行成功

(23)shift.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
(nemu) si 1
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu) q
qemu-system-i386: terminating on signal 15 from pid 15679 ()
 select-sort
tangxi@debian:~/ics2021/nexus-am/tests/cputest$ make ARCH=x86-nemu ALL=shift run
Building shift [x86-nemu]
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/shift-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 1
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

直接运行成功

(24)shuixianhua.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu) q
qemu-system-i386: terminating on signal 15 from pid 15751 (<unknown process>)
 shift
tangxi@debian:~/ics2021/nexus-am/tests/cputest$ make ARCH=x86-nemu ALL=shuixianhua run
Building shuixianhua [x86-nemu]
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/shuixianhua-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) zi 1
Unknown command 'zi'
(nemu) si 1
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

直接运行成功

(25)string.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
(nemu) si 1
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu) q
qemu-system-i386: terminating on signal 15 from pid 15822 ()
 shuixianhua
tangxi@debian:~/ics2021/nexus-am/tests/cputest$ make ARCH=x86-nemu ALL=string run
Building string [x86-nemu]
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/string-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 1
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

直接运行成功

(26)sub-longlong.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
tangxi@debian:~/ics2021/nexus-am/tests/cputest$ make ARCH=x86-nemu ALL=sub-longlong run
Building sub-longlong [x86-nemu]
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/sub-longlong-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 1
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
(nemu) c
invalid opcode(eip = 0x00100085): 1b 55 e4 89 45 f8 89 55 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x00100085 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x00100085) in the disassembling result to distinguish which case it is.

If it is the first case, see
```

1b(sbb)

译码函数 E2G， 执行函数 sbb， sbb 已实现，填表

```
/* 0x18 */ EMPTY, EMPTY, EMPTY, IDEX(E2G, sbb),
```

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  10011c:   89 d8                               movl %ebx,%eax
  10011e:   33 45 d4                            xorl -0x2c(%ebp),%eax
  100121:   89 c7                               movl %eax,%edi
  100123:   89 f0                               movl %esi,%eax
  100125:   09 f8                               orl %edi,%eax
  100127:   85 c0                               testl %eax,%eax
  100129:   0f 94 c0                            sete %al
  10012c:   0f b6 c0                            movzxl %al,%al
  10012f:   83 ec 0c                            subl $0xc,%esp
  100132:   50                                  pushl %eax
  100133:   e8 0a ff ff ff                      call 100042
  100042:   55                                  pushl %ebp
  100043:   89 e5                               movl %esp,%ebp
  100045:   83 ec 08                            subl $0x8,%esp
  100048:   83 7d 08 00                         cmpl $0x0,0x8(%ebp)
  10004c:   75 0d                               jne 10005b
  10005b:   90                                  nop
  10005c:   c9                                  leave
  10005d:   c3                                  ret
  100138:   83 c4 10                            addl $0x10,%esp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

(27)sum.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
tangxi@debian:~/ics2021/nexus-am/tests/cputest$ make ARCH=x86-nemu ALL=sum run
Building sum [x86-nemu]
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/sum-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 9
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
  100005:   bc 00 7c 00 00                      movl $0x7c00,%esp
  10000a:   e8 0f 00 00 00                      call 10001e
  10001e:   55                                  pushl %ebp
  10001f:   89 e5                               movl %esp,%ebp
  100021:   83 ec 18                            subl $0x18,%esp
  100024:   e8 e6 ff ff ff                      call 10000f
  10000f:   55                                  pushl %ebp
  100010:   89 e5                               movl %esp,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

直接运行成功

(28)switch.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
  10000f:   55                                  pushl %ebp
  100010:   89 e5                               movl %esp,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu) q
qemu-system-i386: terminating on signal 15 from pid 16195 (<unknown process>)
 sum
tangxi@debian:~/ics2021/nexus-am/tests/cputest$ make ARCH=x86-nemu ALL=switch run
Building switch [x86-nemu]
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/switch-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 1
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

直接运行成功

(29)to-lower-case.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
(nemu) si 1
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu) q
qemu-system-i386: terminating on signal 15 from pid 16269 ()
 switch
tangxi@debian:~/ics2021/nexus-am/tests/cputest$ make ARCH=x86-nemu ALL=to-lower-case run
Building to-lower-case [x86-nemu]
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/to-lower-case-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 1
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

直接运行成功

(30)unalign.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
(nemu) si 1
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu) q
qemu-system-i386: terminating on signal 15 from pid 16340 (<unknown process>)
 to-lower-case
tangxi@debian:~/ics2021/nexus-am/tests/cputest$ make ARCH=x86-nemu ALL=unalign run
Building unalign [x86-nemu]
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/unalign-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 1
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

直接运行成功

(31) wanshu.c

```
tangxi@debian: ~/ics2021/nexus-am/tests/cputest
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/unalign-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) q
qemu-system-i386: terminating on signal 15 from pid 16482 (<unknown process>)
 unalign
tangxi@debian:~/ics2021/nexus-am/tests/cputest$ make ARCH=x86-nemu ALL=wanshu run
Building wanshu [x86-nemu]
Building am [x86-nemu]
make[2]: *** No targets specified and no makefile found.  Stop.
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/cputest/
build/wanshu-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 15:02:55, Jun  5 2021
For help, type "help"
(nemu) si 1
  100000:   bd 00 00 00 00                      movl $0x0,%ebp
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x0010001b

(nemu)
```

直接运行成功

2.通过一键回归测试

在 nemu/ 目录下运行 bash runall.sh

```
tangxi@debian:~/ics2021/nemu$ bash
NEMU compile OK
compiling testcases...
testcases compile OK
[  add-longlong] PASS!
[           add] PASS!
[           bit] PASS!
[   bubble-sort] PASS!
[         dummy] PASS!
[          fact] PASS!
[           fib] PASS!
[      goldbach] PASS!
[     hello-str] PASS!
[       if-else] PASS!
[     leap-year] PASS!
[    load-store] PASS!
[    matrix-mul] PASS!
[           max] PASS!
[          min3] PASS!
[         mov-c] PASS!
[         movsx] PASS!
[  mul-longlong] PASS!
[        pascal] PASS!
[         prime] PASS!
[    quick-sort] PASS!
[     recursion] PASS!
[   select-sort] PASS!
[         shift] PASS!
[   shuixianhua] PASS!
[        string] PASS!
[  sub-longlong] PASS!
[           sum] PASS!
[        switch] PASS!
[ to-lower-case] PASS!
[       unalign] PASS!
[        wanshu] PASS!
tangxi@debian:~/ics2021/nemu$
```

3.IN/OUT 指令

加入IOE

查询`i386`手册，发现有两个地方需要填表，分别是`e4~e7` `ec~ef`

在 `nemu/include/common.h` 中定义宏 `HAS_IOE`

```
// Define this macro after serial has been implemented
#define HAS_SERIAL
```

实现输入输出指令

`in`:通过`pio_read`读取指定位置的值并写入目的操作数即可。

```
make_EHelper(in) {
  id_dest->val=pio_read(id_src->val,id_dest->width);
  operand_write(id_dest,&id_dest->val);

  print_asm_template2(in);

#ifdef DIFF_TEST
  diff_test_skip_qemu();
#endif
}
```

`out`： 通过`pio_write`读取指定位置的值输出即可。

```
make_EHelper(out) {
  pio_write(id_dest->val,id_dest->width,id_src->val);

  print_asm_template2(out);

#ifdef DIFF_TEST
  diff_test_skip_qemu();
#endif
}
```

并在`all-instr.h`中声明`make_EHelper(in)`和`make_EHelper(out)`

查询`i386`手册，发现有两个地方需要填表，分别是`e4~e7` `ec~ef`

```
  /* 0xe4 */    IDEXW(in_I2a, in, 1), IDEX(in_I2a, in), IDEXW(out_a2I, out, 1),
IDEX(out_a2I, out),
  /* 0xe8 */    IDEX(J, call), IDEX(J, jmp), EMPTY, IDEXW(J,jmp,1),
  /* 0xec */    IDEXW(in_dx2a, in, 1), IDEX(in_dx2a, in), IDEXW(out_a2dx, out,
1), IDEX(out_a2dx, out),
```

然后在`nexus-am/am/arch/x86-nemu/src/trm.c`中定义宏`HAS_SERIAL`

```
// Define this macro after serial has been implemented
#define HAS_SERIAL
```

最后在`nexus-am/apps/hello`目录下输入`make run`

```
tangxi@debian: ~/ics2021/nexus-am/apps/hello
+ CC src/device/vga.c
+ CC src/device/device.c
+ CC src/device/serial.c
+ CC src/device/keyboard.c
+ CC src/device/io/port-io.c
+ CC src/device/io/mmio.c
+ CC src/memory/memory.c
+ CC src/cpu/intr.c
+ CC src/cpu/reg.c
+ CC src/cpu/decode/modrm.c
+ CC src/cpu/decode/decode.c
+ CC src/cpu/exec/logic.c
+ CC src/cpu/exec/system.c
+ CC src/cpu/exec/special.c
+ CC src/cpu/exec/cc.c
+ CC src/cpu/exec/arith.c
+ CC src/cpu/exec/control.c
+ CC src/cpu/exec/exec.c
+ CC src/cpu/exec/prefix.c
+ CC src/cpu/exec/data-mov.c
+ CC src/monitor/cpu-exec.c
+ CC src/monitor/debug/expr.c
+ CC src/monitor/debug/ui.c
+ CC src/monitor/debug/watchpoint.c
+ CC src/monitor/diff-test/protocol.c
+ CC src/monitor/diff-test/diff-test.c
+ CC src/monitor/diff-test/gdb-host.c
+ CC src/monitor/monitor.c
+ LD build/nemu
./build/nemu -l /home/tangxi/ics2021/nexus-am/apps/hello/build/nemu-log.txt /home/tangxi/ics2
021/nexus-am/apps/hello/build/hello-x86-nemu.bin
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/apps/hello/bui
ld/hello-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 02:40:48, Jun  7 2021
For help, type "help"
(nemu) c
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
nemu: HIT GOOD TRAP at eip = 0x001000f1

(nemu)
```

4.实现时钟设备

实现 _uptime() 函数

在 `nexus-am/am/arch/x86-nemu/src/ioe.c` 文件中实现 `unsigned long _uptime()`

```
unsigned long _uptime() {
  return inl(RTC_PORT) - boot_time;
}
```

成功运行 timetest 程序

在 `nexus-am/tests/timetest` 目录下输入make run运行 timetest 程序

```
tangxi@debian: ~/ics2021/nexus-am/tests/timetest
Hello World!
Hello World!
Hello World!
Hello World!
nemu: HIT GOOD TRAP at eip = 0x001000f1

(nemu) q
make[1]: Leaving directory '/home/tangxi/ics2021/nemu'
tangxi@debian:~/ics2021/nexus-am/apps/hello$ qemu-system-i386: terminating on signal 15 from
pid 18776 (<unknown process>)

tangxi@debian:~/ics2021/nexus-am/apps/hello$ cd ../..
tangxi@debian:~/ics2021/nexus-am$ cd tests/timetest/
tangxi@debian:~/ics2021/nexus-am/tests/timetest$ make run
tangxi@debian:~/ics2021/nexus-am/tests/timetest$ make run
Building timetest [x86-nemu]
+ CC main.c
make[1]: Entering directory '/home/tangxi/ics2021/nexus-am'
make[2]: Entering directory '/home/tangxi/ics2021/nexus-am/am'
Building am [x86-nemu]
+ CC arch/x86-nemu/src/ioe.c
+ AR /home/tangxi/ics2021/nexus-am/am/build/am-x86-nemu.a
make[2]: Leaving directory '/home/tangxi/ics2021/nexus-am/am'
make[1]: Leaving directory '/home/tangxi/ics2021/nexus-am'
make[1]: Entering directory '/home/tangxi/ics2021/nexus-am/libs/klib'
make[1]: *** No targets specified and no makefile found.  Stop.
make[1]: Leaving directory '/home/tangxi/ics2021/nexus-am/libs/klib'
make: [/home/tangxi/ics2021/nexus-am/Makefile.compile:86: klib] Error 2 (ignored)
make[1]: Entering directory '/home/tangxi/ics2021/nemu'
./build/nemu -l /home/tangxi/ics2021/nexus-am/tests/timetest/build/nemu-log.txt /home/tangxi/
ics2021/nexus-am/tests/timetest/build/timetest-x86-nemu.bin
[src/monitor/diff-test/diff-test.c,97,init_difftest] Connect to QEMU successfully
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/tests/timetest
/build/timetest-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 02:40:48, Jun  7 2021
For help, type "help"
(nemu) c
1 second.
2 seconds.
3 seconds.
4 seconds.
5 seconds.
6 seconds.
7 seconds.
8 seconds.
9 seconds.
10 seconds.
11 seconds.
12 seconds.
```

5.运行跑分项目

运行 dhrystone , coremark , microbench 三个跑分项目

由于 differential testing 需要与QEMU 进行通信，我关掉了diff来提高运行速度。

注释掉 `nemu/include/common.h` 中的 `DEBUG 和 DIFF_TEST` 宏

```
#define DEBUG
//#define DIFF_TEST
//#define DIFF_TEST
```

dhrystone

运行跑分项目，进入 `nexus-am/apps/dhrystone` 文件中，运行 dhrystone ，发现 3a 操作码对应指令没有实

输入 `make ARCH=x86-native run` 把 dhrystone 编译到 native

```
tangxi@debian:~/ics2021/nexus-am/apps/dhrystone$ make ARCH=x86-native run
/home/tangxi/ics2021/nexus-am/Makefile.check:11: *** Invalid ARCH. Supported: native x86-nemu
.  Stop.
tangxi@debian:~/ics2021/nexus-am/apps/dhrystone$
```

输入`make ARCH=x86-nemu run`等待。

```
/build/nemu -l /home/tangxi/ics2021/nexus-am/apps/dhrystone/build
ics2021/nexus-am/apps/dhrystone/build/dhrystone-x86-nemu.bin
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2
/build/dhrystone-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 03:10:30, Jun  7 20
For help, type "help"
(nemu) c
Dhrystone Benchmark, Version C, Version 2.2
Trying 500000 runs through Dhrystone.
Finished in 40113 ms
==================================================
Dhrystone PASS          25 Marks
                   vs. 100000 Marks (i7-6700 @ 3.40GHz)
nemu: HIT GOOD TRAP at eip = 0x001000f1

(nemu)
```

coremark

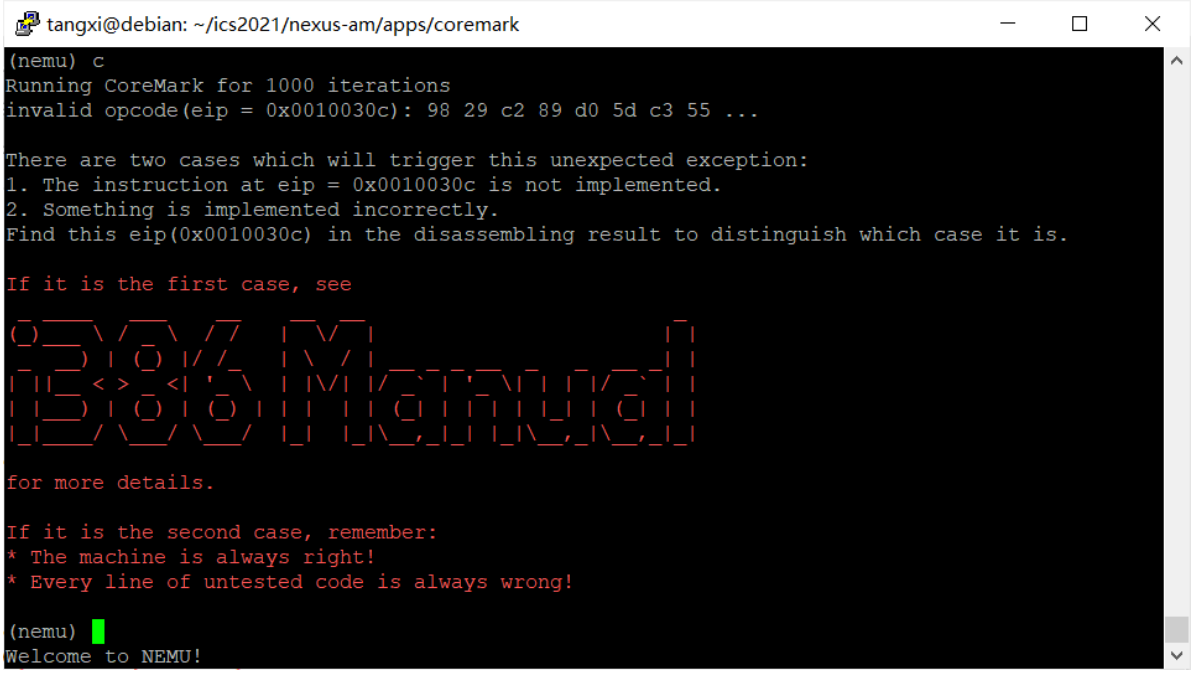
进入 `nexus-am/apps/coremark` 文件中，先用make run运行 coremark

```
tangxi@debian: ~/ics2021/nexus-am/apps/coremark
(nemu) c
Running CoreMark for 1000 iterations
invalid opcode(eip = 0x001006cf): 25 00 07 00 00 66 0b 45 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x001006cf is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x001006cf) in the disassembling result to distinguish which case it is.

If it is the first case, see
 _ ____   ___    __    __ __                          _
(_)___ \ / _ \  / /   |  \/  |                        | |
 _  __) | (_) |/ /_   | \  / | __ _ _ __   _   _ __ _| |
| ||__ < > _ <| '_ \  | |\/| |/ _` | '_ \| | | |/ _` | |
| |___) | (_) | (_) | | |  | | (_| | | | | |_| | (_| | |
|_|____/ \___/ \___/  |_|  |_|\__,_|_| |_|\__,_|\__,_|_|

for more details.

If it is the second case, remember:
* The machine is always right!
* Every line of untested code is always wrong!

(nemu)
Welcome to NEMU!
```

25(and)

译码函数 I2a ，执行函数 and ， and 已实现，填表

```
/* 0x24 */ EMPTY, IDEX(I2a, and), EMPTY, EMPTY,
```



98(cwtl)

无译码函数，执行函数 cwtl ， 填表

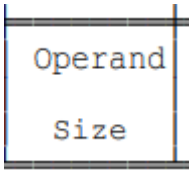
```
/* 0x98 */ EX(cwtl), EX(cltd), EMPTY, EMPTY,
```

在 `data-mov.c` 中完成 `make_EHelper_(cwtl)`

```
make_EHelper(cwtl) {
  if (decoding.is_operand_size_16) {
    rtl_lr_b(&t0, R_AX);
    rtl_sext(&t0, &t0, 1);
    rtl_sr_w(R_AX, &t0);
  }
  else {
    rtl_lr_w(&t0, R_AX);
    rtl_sext(&t0, &t0, 2);
    rtl_sr_l(R_EAX, &t0);
  }

  print_asm(decoding.is_operand_size_16 ? "cbtw" : "cwtl");
}
```

```
tangxi@debian: ~/ics2021/nexus-am/apps/coremark
(nemu) c
Running CoreMark for 1000 iterations
invalid opcode(eip = 0x00101bc5): 66 81 4d e8 00 80 eb 06 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x00101bc5 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x00101bc5) in the disassembling result to distinguish which case it is.

If it is the first case, see

for more details.

If it is the second case, remember:
* The machine is always right!
* Every line of untested code is always wrong!

(nemu)
Welcome to NEMU!
```

66 81(or)



66 对应 EX(operand_size)，这是我们需实现 66 后一位的指令，也就是 81

```
/* 0x80 */ IDEXW(I2E, gp1, 1), IDEX(I2E, gp1), EMPTY, IDEX(SI2E,
gp1),
```

查看反汇编可知需实现 or 指令，查阅i386手册可知 or 对应 001 项， make_EHelper(or) 已 实现，填表

```
make_group(gp1,
EX(add), EX(or), EMPTY, EMPTY,
EX(and), EX(sub), EX(xor), EX(cmp))
```

```
tangxi@debian: ~/ics2021/nexus-am/apps/coremark
(nemu) c
Running CoreMark for 1000 iterations
invalid opcode(eip = 0x00100262): 0c 80 66 0b 45 e6 8b 55 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x00100262 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x00100262) in the disassembling result to distinguish which case it is.

If it is the first case, see

for more details.

If it is the second case, remember:
* The machine is always right!
* Every line of untested code is always wrong!

(nemu)
Welcome to NEMU!
```

0c(or),0d(0r)

译码函数 I2a ， 执行函数 or ， 宽度1， or 已实现，填表

```
/* 0x0c */ IDEXW(I2a, or, 1), IDEX(I2a, or), EMPTY, EX(2byte_esc),
```

tangxi@debian: ~/ics2021/nexus-am/apps/coremark

```
(nemu) c
Running CoreMark for 1000 iterations
invalid opcode(eip = 0x00102079): 0f 9f c0 0f b6 d0 66 8b ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x00102079 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x00102079) in the disassembling result to distinguish which case it is.

If it is the first case, see
 _ ____   ___    __    __ __                          _
(_)___ \ / _ \  / /   |  \/  |                        | |
 _   __) | (_) |/ /_   | \  / | __ _ _ __   _   _  __ _| |
| ||__ < > _ <| '_ \  | |\/| |/ _` | '_ \| | | |/ _` | |
| |___) | (_) | (_) | | |  | | (_| | | | | |_| | (_| | |
|_|____/ \___/ \___/  |_|  |_|\__,_|_| |_|\__,_|\__,_|_|

for more details.

If it is the second case, remember:
* The machine is always right!
* Every line of untested code is always wrong!

(nemu)
Welcome to NEMU!
```

0f 9f(set)

译码函数 E ， 执行函数 setcc ， setcc 已实现，填完90~9f

```
/* 0x90 */ IDEXW(E, setcc, 1), IDEXW(E, setcc, 1), IDEXW(E, setcc, 1), IDEXW(E,
setcc, 1),
/* 0x94 */ IDEXW(E, setcc, 1), IDEXW(E, setcc, 1), IDEXW(E, setcc, 1), IDEXW(E,
setcc, 1),
/* 0x98 */ IDEXW(E, setcc, 1), IDEXW(E, setcc, 1), IDEXW(E, setcc, 1), IDEXW(E,
setcc, 1),
/* 0x9c */ IDEXW(E, setcc, 1), IDEXW(E, setcc, 1), IDEXW(E, setcc, 1), IDEXW(E,
setcc, 1),
```

tangxi@debian: ~/ics2021/nexus-am/apps/coremark

```
./build/nemu -l /home/tangxi/ics2021/nexus-am/apps/coremark/build/nemu-log.txt /home/tangxi/i
cs2021/nexus-am/apps/coremark/build/coremark-x86-nemu.bin
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/apps/coremark/
build/coremark-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 03:10:30, Jun  7 2021
For help, type "help"
(nemu) c
Running CoreMark for 1000 iterations
CoreMark Size    : 666
Total time (ms)  : 91599
Iterations       : 1000
Compiler version : GCC8.3.0
seedcrc          : 0xfffd
[0]crclist       : 0xa5ad
[0]crcmatrix     : 0x000a
[0]crcstate      : 0xffff
[0]crcfinal      : 0xa5ad
Finised in 91599 ms.
Cannot validate operation for these seed values, please compare with results on a known platf
orm.
nemu: HIT GOOD TRAP at eip = 0x001000f1

(nemu)
```

用`x86-nemu`跑，输入`make ARCH=native run 把 coremark`编译到native后，输入`make ARCH=x86-nemu run`

```
tangxi@debian: ~/ics2021/nexus-am/apps/coremark
Building am [native]
make[2]: Nothing to be done for 'archive'.
make[2]: Leaving directory '/home/tangxi/ics2021/nexus-am/am'
make[1]: Leaving directory '/home/tangxi/ics2021/nexus-am'
make[1]: Entering directory '/home/tangxi/ics2021/nexus-am/libs/klib'
make[1]: *** No targets specified and no makefile found.  Stop.
make[1]: Leaving directory '/home/tangxi/ics2021/nexus-am/libs/klib'
make: [/home/tangxi/ics2021/nexus-am/Makefile.compile:86: klib] Error 2 (ignored)
Running CoreMark for 1000 iterations
2K performance run parameters for coremark.
CoreMark Size    : 666
Total time (ms)  : 208
Iterations       : 1000
Compiler version : GCC8.3.0
seedcrc          : 0xe9f5
[0]crclist       : 0xe714
[0]crcmatrix     : 0x1fd7
[0]crcstate      : 0x8e3a
[0]crcfinal      : 0xd340
Finished in 208 ms.
==================================================
CoreMark PASS       21483 Marks
                vs. 100000 Marks (i7-6700 @ 3.40GHz)
tangxi@debian:~/ics2021/nexus-am/apps/coremark$
```

```
tangxi@debian: ~/ics2021/nexus-am/apps/coremark
./build/nemu -l /home/tangxi/ics2021/nexus-am/apps/coremark/build/nemu-log.txt /home/tangxi/i
cs2021/nexus-am/apps/coremark/build/coremark-x86-nemu.bin
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/apps/coremark/
build/coremark-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 03:10:30, Jun  7 2021
For help, type "help"
(nemu) c
Running CoreMark for 1000 iterations
CoreMark Size    : 666
Total time (ms)  : 91325
Iterations       : 1000
Compiler version : GCC8.3.0
seedcrc          : 0xfffd
[0]crclist       : 0xa5ad
[0]crcmatrix     : 0x000a
[0]crcstate      : 0xffff
[0]crcfinal      : 0xa5ad
Finished in 91325 ms.
Cannot validate operation for these seed values, please compare with results on a known platf
orm.
nemu: HIT GOOD TRAP at eip = 0x001000f1

(nemu)
```

microbench

进入`nexus-am/apps/microbench/`目录中，运行microbench，发现有指令没有实现 23(and)

```
tangxi@debian: ~/ics2021/nexus-am/apps/microbench
[qsort] Quick sort: * Passed.
  min time: 5128 ms [107]
[queen] Queen placement: invalid opcode(eip = 0x00103443): 23 45 f4 89 45 ec 8b 45 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x00103443 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x00103443) in the disassembling result to distinguish which case it is.

If it is the first case, see

 _ ____   ___    __    __ __                        _
(_)___ \ / _ \  / /   |  \/  |                      | |
 _  __) | (_) |/ /_   | \  / | __ _ _ __  _   _  __ _| |
| ||__ < > _ <| '_ \  | |\/| |/ _` | '_ \| | | |/ _` | |
| |___) | (_) | (_) | | |  | | (_| | | | | |_| | (_| | |
|_|____/ \___/ \___/  |_|  |_|\__,_|_| |_|\__,_|\__,_|_|

for more details.

If it is the second case, remember:
* The machine is always right!
* Every line of untested code is always wrong!

(nemu)
```

23(and)

译码函数 E2G ， 执行函数 and ， and 已实现，填表

```
/* 0x20 */ EMPTY, IDEX(G2E, and), IDEXW(E2G, and, 1), IDEX(G2E and),
```

这个忘记截图了

c2(reti)

译码函数 I ， 执行函数 reti ， 宽度2，填表

```
/* 0xc0 */ IDEXW(gp2_Ib2E, gp2, 1), IDEX(gp2_Ib2E, gp2), IDEXW(I, reti, 2),
EX(ret),
```

```
tangxi@debian: ~/ics2021/nexus-am/apps/microbench
+ LD build/nemu
./build/nemu -l /home/tangxi/ics2021/nexus-am/apps/microbench/build/nemu-log.txt /home/tangxi
/ics2021/nexus-am/apps/microbench/build/microbench-x86-nemu.bin
[src/monitor/monitor.c,65,load_img] The image is /home/tangxi/ics2021/nexus-am/apps/microbenc
h/build/microbench-x86-nemu.bin
PuTTY X11 proxy: unable to connect to forwarded X server: Network error: Connection refused
PuTTY X11 proxy: unable to connect to forwarded X server: Network error: Connection refused
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 07:39:47, Jun  7 2021
For help, type "help"
(nemu) c
[qsort] Quick sort: * Passed.
  min time: 6826 ms [80]
[queen] Queen placement: * Passed.
  min time: 4470 ms [115]
[bf] Brainf**k interpreter: * Passed.
  min time: 44511 ms [58]
[fib] Fibonacci number: * Passed.
  min time: 413296 ms [6]
[sieve] Eratosthenes sieve: * Passed.
  min time: 140635 ms [30]
[15pz] A* 15-puzzle search: Assertion fail at src/15pz/15pz.cpp:37
nemu: HIT BAD TRAP at eip = 0x001000f1
```

输入 `make ARCH=native run` 把microbench 编译到 native

```
tangxi@debian: ~/ics2021/nexus-am/apps/microbench
  min time: 20 ms [27595]
[queen] Queen placement: * Passed.
  min time: 13 ms [39684]
[bf] Brainf**k interpreter: * Passed.
  min time: 151 ms [17356]
[fib] Fibonacci number: * Passed.
  min time: 1180 ms [2421]
[sieve] Eratosthenes sieve: * Passed.
  min time: 340 ms [12472]
[15pz] A* 15-puzzle search: * Passed.
  min time: 41 ms [14126]
[dinic] Dinic's maxflow algorithm: * Passed.
  min time: 23 ms [58852]
[lzip] Lzip compression: * Passed.
  min time: 114 ms [23218]
[ssort] Suffix sort: * Passed.
  min time: 14 ms [42250]
[md5] MD5 digest: * Passed.
  min time: 62 ms [31601]
====================================================
MicroBench PASS        26957 Marks
                  vs. 100000 Marks (i7-6700 @ 3.40GHz)
Exit (0)
tangxi@debian:~/ics2021/nexus-am/apps/microbench$
```

输入 `make ARCH=x86-nemu run`

```
(nemu) c
[qsort] Quick sort: * Passed.
  min time: 5242 ms [105]
[queen] Queen placement: * Passed.
  min time: 3485 ms [148]
[bf] Brainf**k interpreter: * Passed.
  min time: 35895 ms [73]
[fib] Fibonacci number: * Passed.
  min time: 335456 ms [8]
[sieve] Eratosthenes sieve: * Passed.
  min time: 119239 ms [35]
[15pz] A* 15-puzzle search: * Passed.
  min time: 20408 ms [28]
[dinic] Dinic's maxflow algorithm: * Passed.
  min time: 10518 ms [128]
[lzip] Lzip compression: * Passed.
  min time: 62201 ms [42]
[ssort] Suffix sort: * Passed.
  min time: 6483 ms [91]
[md5] MD5 digest: * Passed.
  min time: 54792 ms [35]
====================================================
MicroBench PASS        69 Marks
                  vs. 100000 Marks (i7-6700 @ 3.40GHz)
nemu: HIT GOOD TRAP at eip = 0x001000f1
```

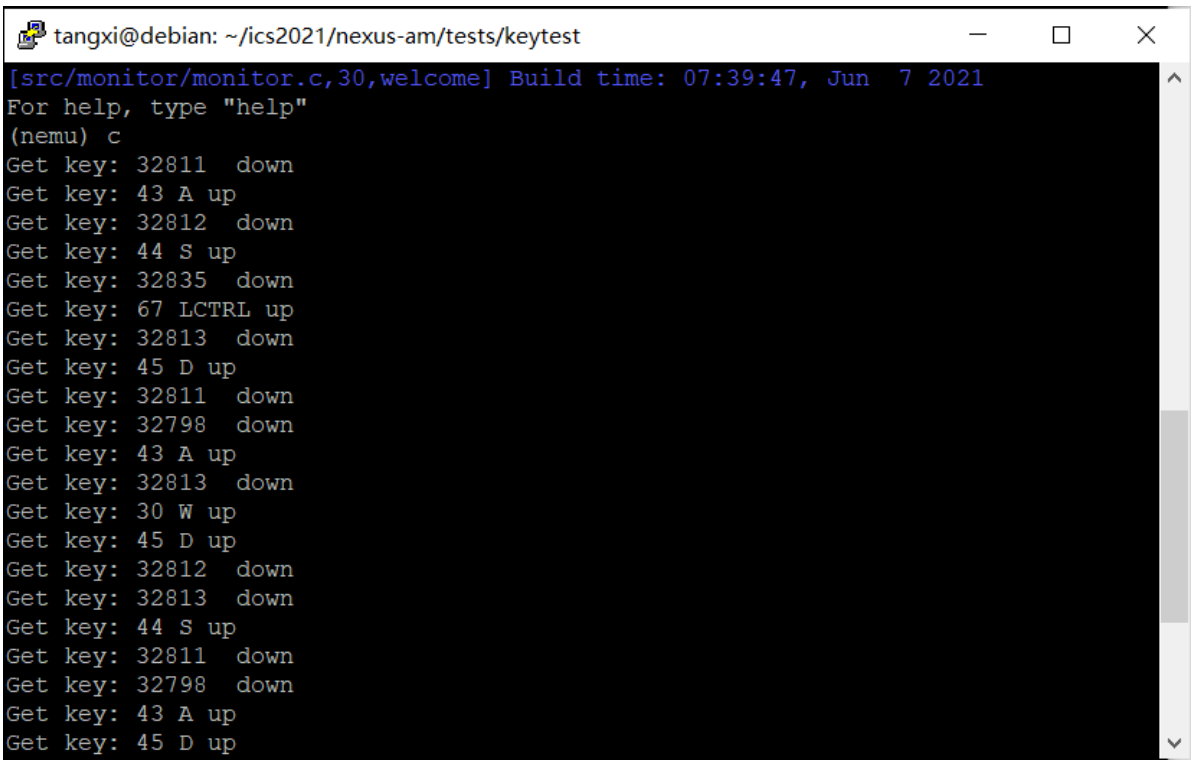
6.实现键盘设备

实现 _read_key() 函数

进入 `nexus-am/am/arch/x86-nemu/src/ioe.c` 中实现 int _read_key()

```
int _read_key() {
    uint32_t key_code = _KEY_NONE;
    if (inb(0x64) & 0x1)
    key_code = inl(0x60);
    return key_code;
}
```

打开xming应用，退出重新登陆putty，在SSH下勾选Xming，然后在 `nexus-am/tests/keytest` 目录下输入make run运行 keytest 程序



成功运行 keytest 程序

7.添加内存映射 I/O

在 paddr_read() 和 paddr_write() 中添加内存映射 I/O 判断

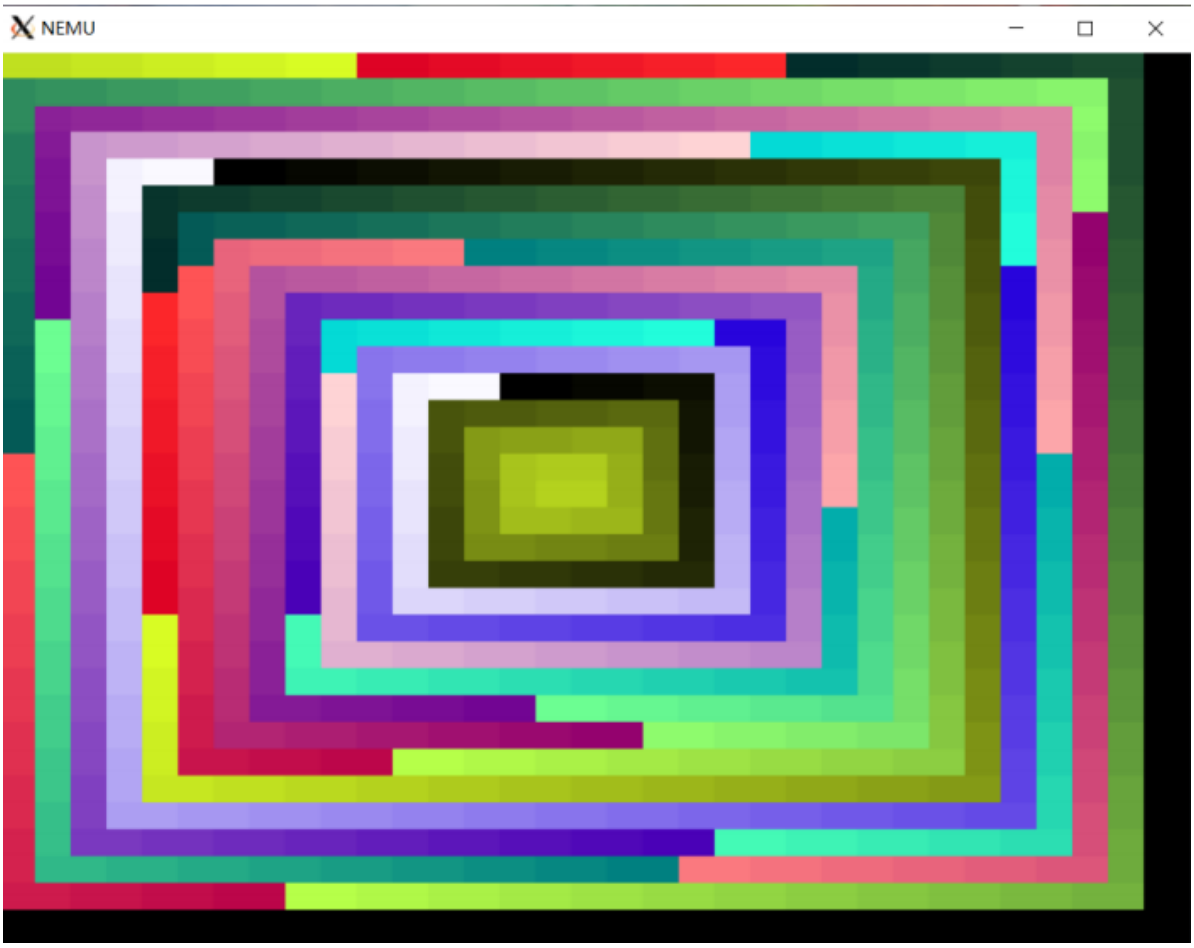
先用`is_mmio`判断一个物理地址是否被映射到 I/O 空间如果是， `is_mmio()` 会返回映射号。 否则返回 `-1`.。内存映射 I/O 的访问需要调用 `mmio_read()` 或 `mmio_write()`，，调用时需要提供映射号。 如果不是内存映射 I/O 的访问， 就访问 `pmem`。

```
uint32_t paddr_read(paddr_t addr, int len) {
    int mmio_id = is_mmio(addr);
    if (mmio_id != -1) {
      return mmio_read(addr, len, mmio_id);
}
  return pmem_rw(addr, uint32_t) & (~0u >> ((4 - len) << 3));
}

void paddr_write(paddr_t addr, int len, uint32_t data) {
   int mmio_id = is_mmio(addr);
```

```
    if (mmio_id != -1) {
        mmio_write(addr, len, data, mmio_id);
    }
    else {
        memcpy(guest_to_host(addr), &data, len);
    }
}
```

成功运行 videotest 程序

进入`nexus-am/tests/videotest`输入make run



8.运行打字小游戏

帧数（FPS）不低于 3

在 `nexus-am/apps/typing` 目录下运行

```
tangxi@debian: ~/ics2021/nexus-am/apps/typing
For help, type "help"
(nemu) c
invalid opcode(eip = 0x00100849): 2d 40 01 00 00 89 c2 c1 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x00100849 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x00100849) in the disassembling result to distinguish which case it is.

If it is the first case, see

 _ ____   ___    __    __ __                        _
(_)___ \ / _ \  / /   |  \/  |                      | |
 _   __) | (_) |/ /_   | \  / | __ _ _ __  _   _  __ _| |
| ||__ < > _ <| '_ \  | |\/| |/ _` | '_ \| | | |/ _` | |
| |___) | (_) | (_) | | |  | | (_| | | | | |_| | (_| | |
|_|____/ \___/ \___/  |_|  |_|\__,_|_| |_|\__,_|\__,_|_|

for more details.

If it is the second case, remember:
* The machine is always right!
* Every line of untested code is always wrong!

(nemu)
asyetest  cputest  floattest  fuzz  guitest  keytest  timetest  videotest
```

发现有指令没有实现，查看2b发现是sub，

2b(sub)

译码函数 I2A ， 执行函数 sub ， sub 已实现，填表

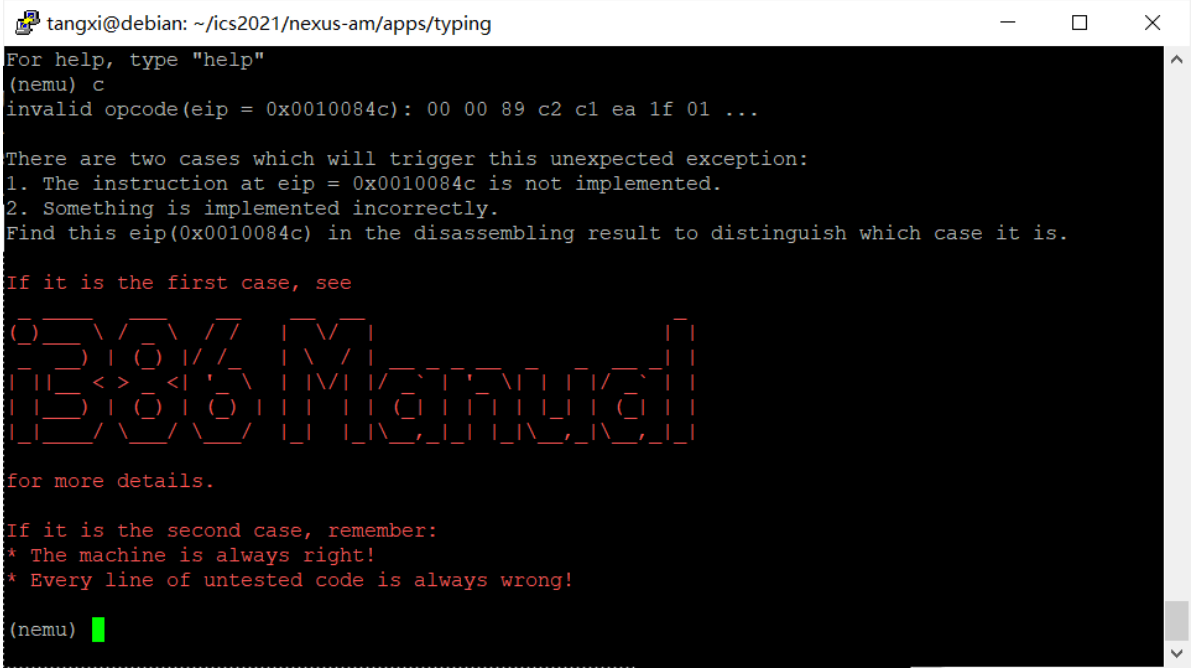
```
/* 0x2c */  EMPTY, IDEX(I2a, sub), EMPTY, EMPTY,
```

```
tangxi@debian: ~/ics2021/nexus-am/apps/typing
For help, type "help"
(nemu) c
invalid opcode(eip = 0x00101185): 29 c2 89 d0 c9 c3 55 89 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x00101185 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x00101185) in the disassembling result to distinguish which case it is.

If it is the first case, see

 _ ____   ___    __    __ __                        _
(_)___ \ / _ \  / /   |  \/  |                      | |
 _   __) | (_) |/ /_   | \  / | __ _ _ __  _   _  __ _| |
| ||__ < > _ <| '_ \  | |\/| |/ _` | '_ \| | | |/ _` | |
| |___) | (_) | (_) | | |  | | (_| | | | | |_| | (_| | |
|_|____/ \___/ \___/  |_|  |_|\__,_|_| |_|\__,_|\__,_|_|

for more details.

If it is the second case, remember:
* The machine is always right!
* Every line of untested code is always wrong!

(nemu)
```

29(sub)

译码函数 G2E ， 执行函数 sub ， sub 已实现，填表

```
/* 0x28 */ EMPTY, IDEX(G2E, sub), EMPTY, IDEX(E2G, sub),
```



00(add)

译码函数 E2G， 执行函数 add， add 已实现，填表

```
/* 0x00 */  IDEXW(G2E, add, 1), IDEX(G2E, add), EMPTY, EMPTY,
```

9.捕捉死循环

TODO();

遇到的问题和解决办法

重大问题：
在add.c和add-longlong.c测试时，文件链接时重定位因为未知的问题导致地址错误，反汇编代码和测试用例的txt文件不一致，所以没办法通过查看反汇编来找到相应的代码，遇到i386表中和grp相关的指令就只能全部一个个地实现，第一个测试也用了大部分的时间，完成了大量的指令，甚至不在用例里面的也写了。

跑分的时候显示地址越界，根据PA1.1的内容，去memory.c里面拓展addr的范围到1024^3即可。

跑分第三个项目microbench最后是bad trap，目前还没解决。

打字小游戏最后显示在右下角，没有居中。

实验心得

通过这次的实验我进一步学会了如何查阅i386手册来选择、实现指令的译码函数和执行函数，并且在思考题中对计算机的输入与输出，cpu的作用有了更深刻的认识。

其他备注

无