方法1：装32位兼容库，<http://www.cnblogs.com/leaven/p/5084902.html>

然后，在sim-outorder目录里，make即可

方法2：编译为64位程序，如下：

命令 cd  sim-outoder所在的目录:  
1. 改Makefile, optical ... / makefile, popnet.../ Makefile, popnet.../ orion .../power/Makefile  : 删去 -m32,  
2. 重新编译64位cacti: gcc \*.c -c, ar rv libcatic.a \*.o  
3. make clean for all the directory. Then make !!!

如果这一步make不出错就ok了。如果出错，继续4-5步  
4. in popnet.../ sim\_router.h, comment flit\_state\_[a].erase(flit\_state\_[a].begin());  
5. in popnet.../ mess\_queue.cc, add #include <algorithm>, if it complains about the find (L.begin(), L.end(), msgNo) == L.end() cannot find a template.

6. add the following sentences to accept\_flit (sim\_router.cc), right before the write output file ends in router (0,0).  判断dest为0 0 并写出输出文件的语句块之内。

                        flock(fp\_b->\_fileno, LOCK\_UN);

}

+delay\_update(t);

+return 1;

}

}

cmp0和cmp1都这么改。

7. 如果仅仅跑popnet only， 可以用 ~/CMP\_WTF/wangling/popnetmodified\_ok/popnet。如果把Makefile 和 orion .../power/Makefile中的-O选项去掉，即只留-g调试选项，则会跑不下去，seg fault。用gdb跟踪，发现是power\_buffer\_write函数(sim\_router.cc中)中某一行i越界。因为这里面是orion的函数，用了FUNC宏定义，非常复杂，无法跟踪。所以原因:FUNC的写函数时，超过了数组的大小，直接将stack变量i给覆盖了。解决方法，在power\_buffer\_write函数之上定义一个int i，全局变量。在power\_buffer\_write函数内，用这个i。即：

+int i = 0;

void power\_template::power\_buffer\_write(long in\_port, Data\_type & write\_d)

{

for( <<去掉int>>i = 0; i < flit\_size\_; i ++) {

8. 如果报错找不到Orion里面的函数，那么找到那个文件，把前面inline去掉