使用方法，点击Q盘映射。

测试方案，在核0上创建一个timer，定时post一个信号量，启动消息传递。

task\_create用于在每个核上都创建两个不同的消息队列。

下面提到的从 task\_malloc\_syn — task\_pend\_syn使用的消息队列的名字是core#to Syn，

从task\_malloc\_nor—task\_pend\_nor使用的消息队列的名字是core#to Nor。

（#表示核号，比如core0to Syn表示核0上的消息队列）

核0上创建一个任务task\_malloc\_syn，在pend到由timer所post的信号量后，给每个核的task\_pend\_syn（包括核0）发送一个消息，通知各个核启动消息的传递。

每个核的task\_pend\_syn在收到消息后，会给各自核上的一个task\_malloc\_nor去post一个信号量，各个核的task\_malloc\_nor在pend到这个信号量后，会给每一个核都发出一条消息，消息的内容为：核# to 核#，比如1 to 2，表示消息是从核1发到核2的。各个核的task\_pend\_nor用来接收由task\_malloc\_nor发来的消息，并对消息的内容进行判断。

如果程序可以正常的执行的话，每一轮的消息传递应该可以打印出16条消息，0-0,0-1,0-2,0-3类似的等等。

在没有开启cache的情况下，消息的传递时正确的。

在L2cache配置成128K，L1D配置成32K的情况下，前面几轮的消息传递时正常的，消息的数目和内容都是对的，但是到了后面，要么消息传递出现了错误，要么消息却越传越少。