

** **HUNAN UNIVERSITY**

路由与交换机

实验报告

|  |  |  |
| --- | --- | --- |
|  | 小组成员： | 计科1802张继伟，谢正宇 |

目录

[一、 实验目的 3](#_Toc67435325)

[二、 实验内容 3](#_Toc67435326)

[三、 实验环境 3](#_Toc67435327)

[2进3出端口配置 4](#_Toc67435328)

[四、 项目1实验步骤 4](#_Toc67435329)

[N进N出端口配置 8](#_Toc67435330)

[五、 项目2实验步骤 8](#_Toc67435331)

[六、 实验思考 10](#_Toc67435332)

[七、 实验思考（个人部分单独完成） 10](#_Toc67435333)

# 实验一

1. 实验目的

1、熟悉NetMagic08的硬件编程方式；

2、基于 NetMagic08 搭建实验环境，包括 NetMagic08 的安装、Quartus 与 NetMagic08的对接

3、使用 Quartus设计硬件逻辑

4、了解FPGA编程基础

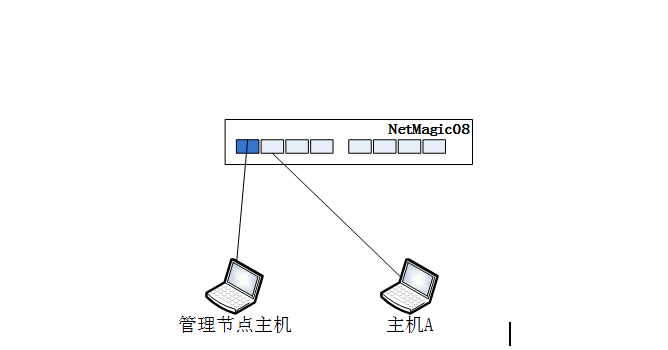
1. 实验内容
2. 基础要求：

在UM\_my/UM.v中使用Verilog语言编写一个模块让NetMagic08实现2口进3口出的基本功能。

1. 扩展要求：

在实现了实验内容1后，改写程序实现从NetMagic08 的2端口进入的包转发至所有端口；所有端口进入的包都转发到2号端口。

1. 实验环境



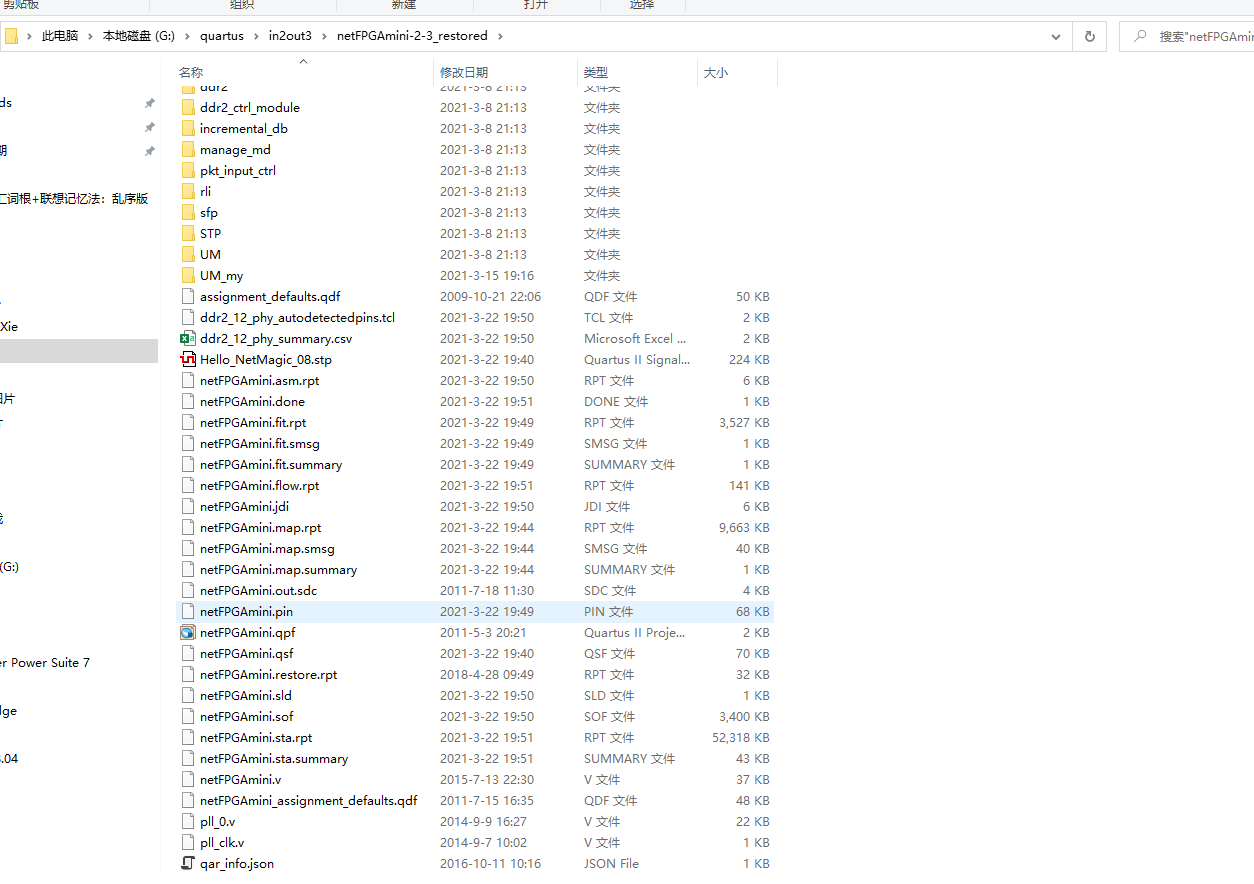
* + - 1. 1台管理节点主机；1台主机A；（分别连接到2口和3口）；
      2. 2根网线；
      3. NetMagic08开发平台；
      4. 软件Quartus 16。主机及网络详细配置参照附带的实验环境拓扑及软件配置文档。

# 2进3出端口配置

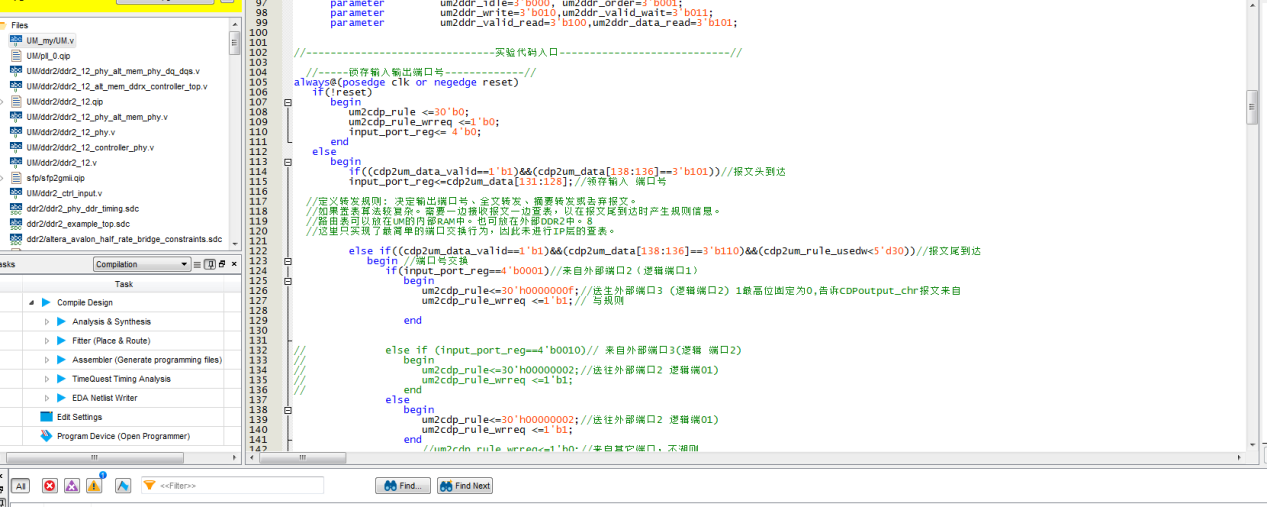
1. 项目1实验步骤

我们首先完成基础要求，实现2进3出的功能配置。

step1．使用Quartus 打开 in2out3目录下的硬件工程项目netFPGAmini-2-3.qsf。

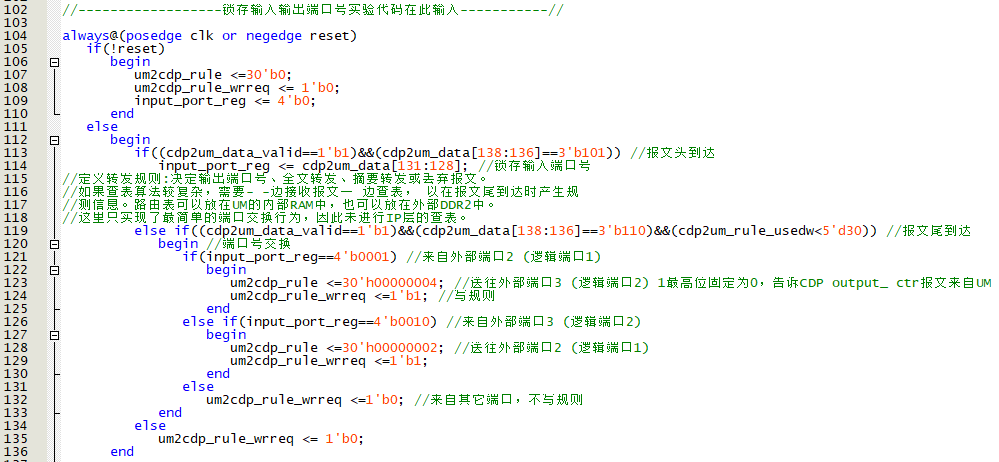


step2．在工程项目中，打开文件列表对UM.v文件进行编写（用户模块文件）

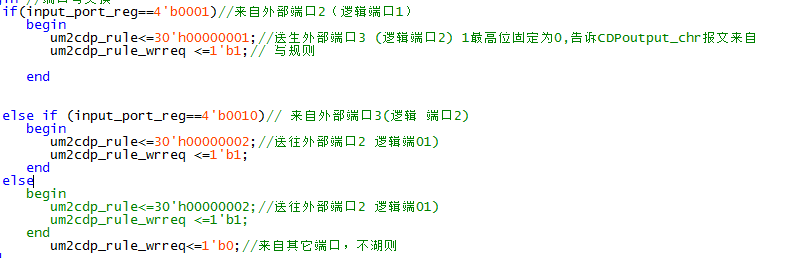


Step3． 在UM.v文件中增加一个新的处理逻辑，在一个always@(posedge clk or negedge reset)逻辑中完成端口2进端口3出的硬件设计。

相关代码如下图：



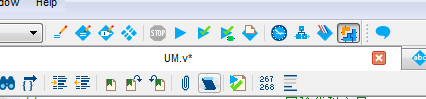
Step4.关键代码解释：



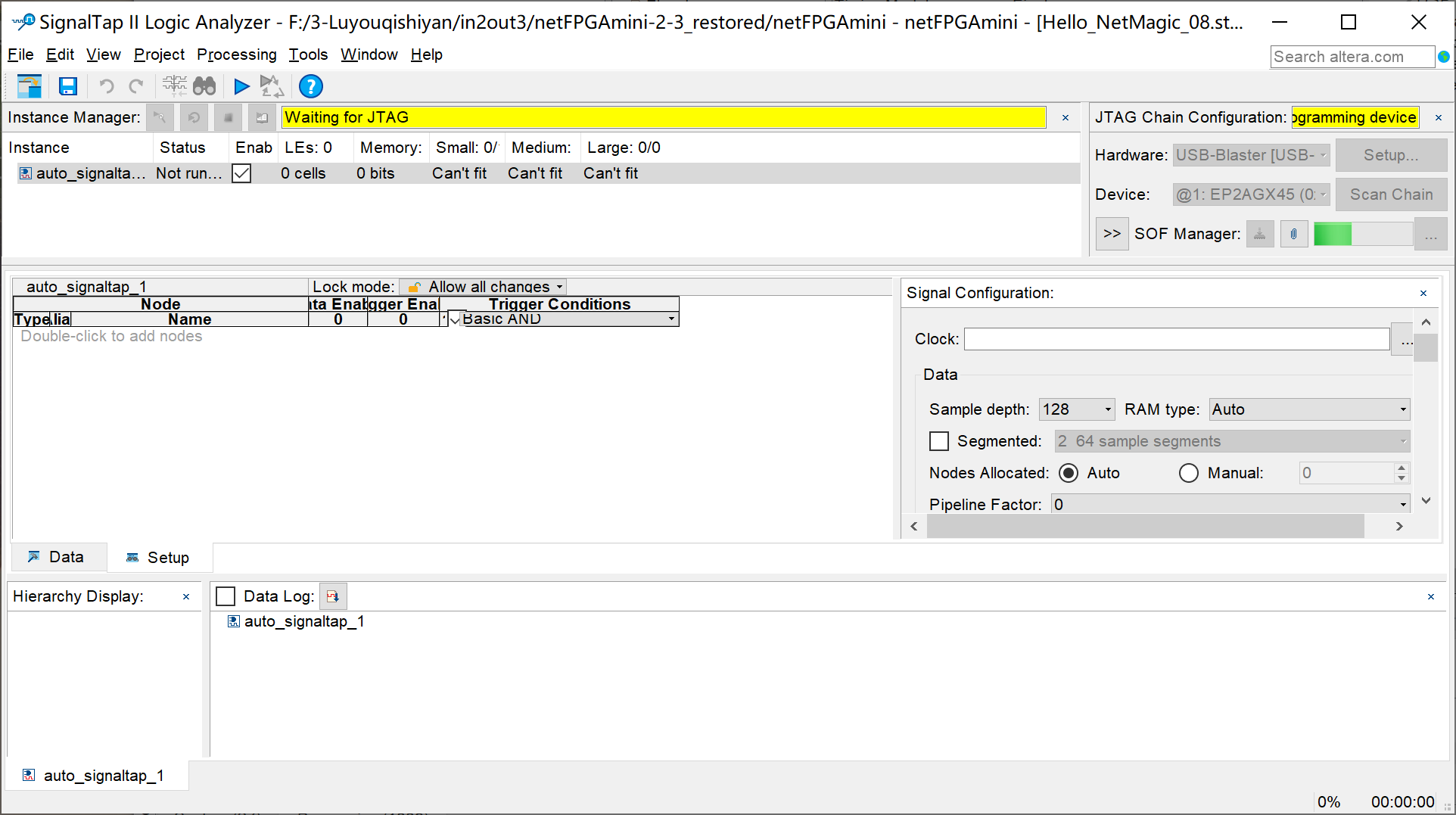
主要的就是内部的if判断语句首先判断输入信息是否是来自端口2的，如果是来自端口2则将信息转发到端口3；同理判断输入信息是否来自端口3，若果是来自端口3则将信息转发到端口2；最后，对于其他情况，我们不转发。

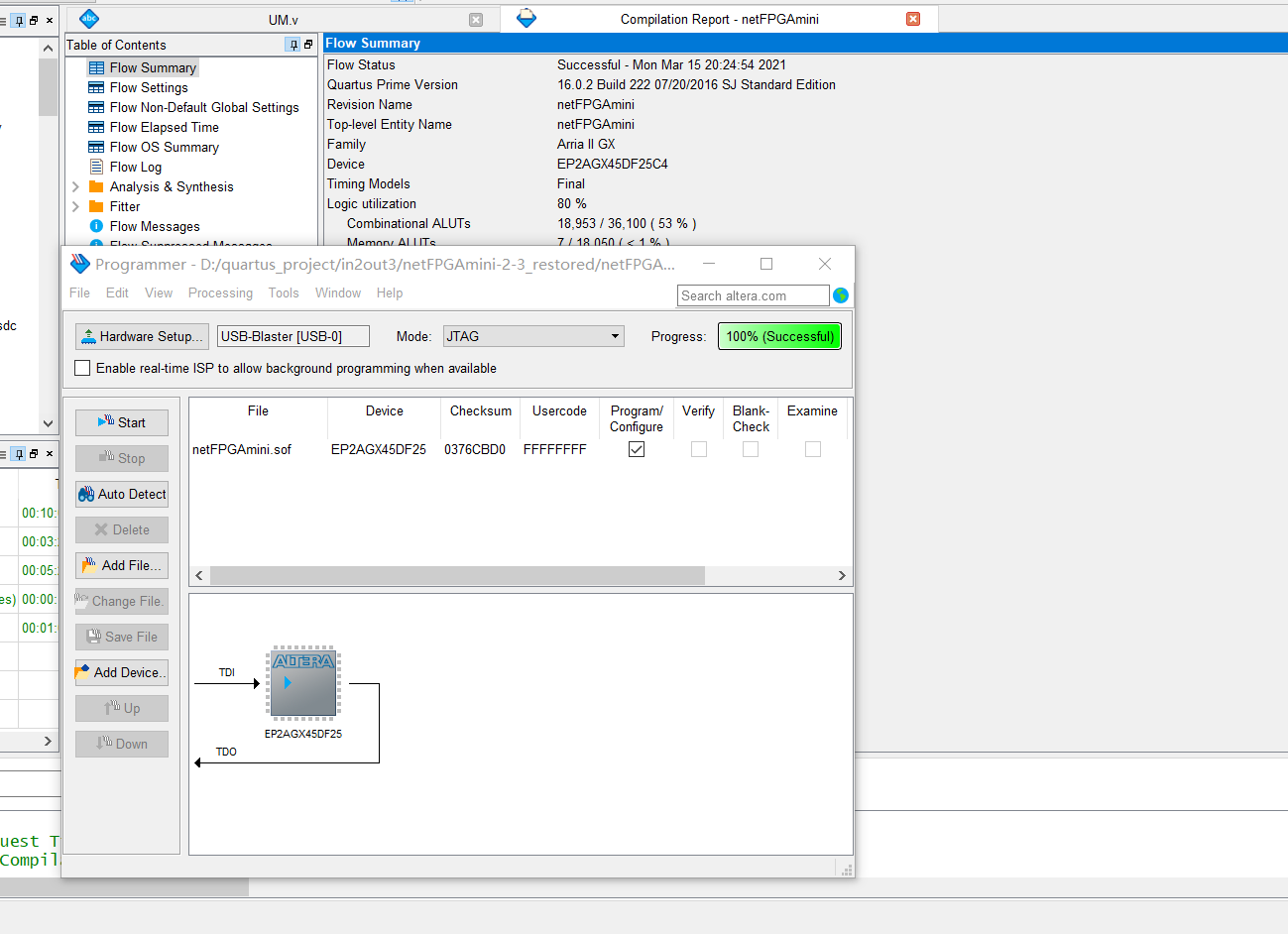
Step5．点击编译命令，对代码进行编译，软件会完成硬件电路的设计和优化，生成和工程名同名的sof文件。

点击图中的蓝色箭头即可进行编译。编译时间可能会较长。



Step6．将sof文件下载到NetMagic08中进行硬件调试



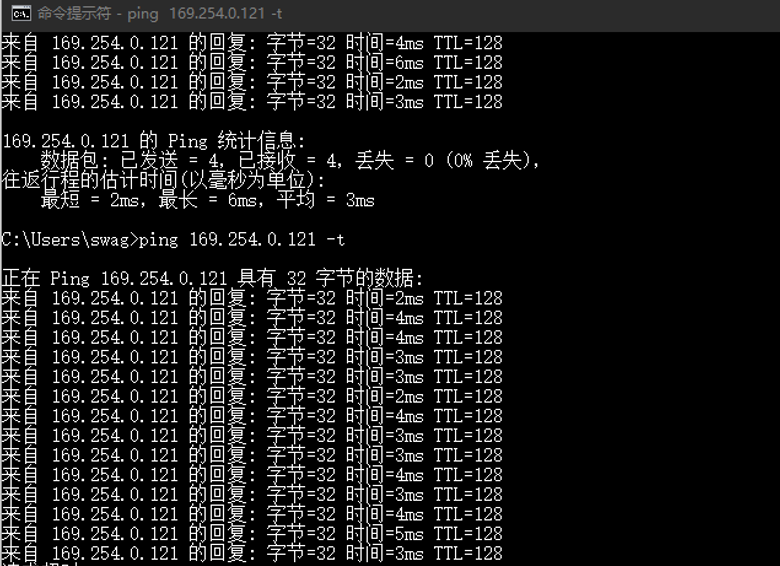


下载完毕后，我们使用网线将两台PC与NetMagic设备相连接，在设置了各自的静态IP地址后，使用命令行名ping命令，查看是否连通。

我们将两台计算机的一台IP设置为：169.254.0.121；另一台设置为：169.254.141.235



从169.254.141.235顺利ping通169.254.0.121（此时169.254.141.235连接2口，169.254.0.121连接3口）



再从169.254.141.235ping169.254.0.121，发现顺利ping通。

断开网口，发现请求超时

至此项目实现2进3出的代码顺利调试成功。

# N进N出端口配置

在完成了项目1中的2进3出的代码后，我们追求更进一步的配置功能。实现N进N出，即实现N号端口连通到其他所有3个端口（当然也可以是其他某几个端口）

1. 项目2实验步骤

修改代码如下：



可以看到，主要是对端口号if判断分支进行了修改。

首先，从2号端口进来的数据（b0001，二进制的1）将会被发送到1、3、4三个端口（h0000000d，最后一位相当于二进制的1101），这样，只要能够连接到相应的目的主机，就能实现连接。

同理，从3号端口进来的数据（b0010，二进制的2）将会被发送到1、2、4三个端口（h0000000b，最后一位相当于二进制的1011），这样，只要能够连接到相应的目的主机，就能实现连接。

同理，从4号端口进来的数据（b0011，二进制的3）将会被发送到1、2、3三个端口（h00000007，最后一位相当于二进制的0111），这样，只要能够连接到相应的目的主机，就能实现连接。

同理，从1号端口进来的数据（b0000，二进制的0）将会被发送到2、3、4三个端口（h0000000e，最后一位相当于二进制的1110），这样，只要能够连接到相应的目的主机，就能实现连接。

相当于我这里实现的是一个广播，任何两个口之间都可以实现互联（项目2验收要求）。后四位从左到右（从高到低）依次代表4、3、2、1号端口。

将网线分别插入2，3号端口：



实验成功，证明此时2号和3号端口能成功连通。

现在我们将线连接在1号端口和4号端口：

再在两台PC机上使用ping命令进行连通：



中间有丢包是以为中途拔掉了网线，然后又插上了。

至此实验已经可以证明成功。

成功实现n进n出的进阶实验。

1. 实验思考

项目2完成后3号口和4号口能否联通？

答案是能。因为代码内部的if判断语句判断的是输入信息是否是来自端口N的，如果是则将信息转发到其他端口，也就是说只要是两个不同端口就可以进行联通。

1. 实验思考（个人部分单独完成）

2进3出的设计特别简单，它只能从2号端口进入，3号端口传出，而对其他端口的使用并不会产生传输结果，我们在2进3出的基础上做出了n进n出的逻辑设计，改动了判断信息来源和转发这一部分的关键代码，能够做到任意一个端口和其他三个端口的互通。单播的信息的接收和传递只在两个节点之间进行，如互相ping时抓取的包，包的source和destination分别是两台pc机。单播在网络中得到了广泛的应用，网络上绝大部分的数据都是以单播的形式传输的，只是一般网络用户不知道而已。例如，你在收发电子邮件、浏览网页时，必须与邮件服务器、Web服务器建立连接，此时使用的就是单播数据传输方式。而广播，发出点是R1，目的地址是ff:ff:ff:ff:ff:ff，也就是所有的地址。广播是主机之间“一对所有”的通讯模式，网络对其中每一台主机发出的信号都进行无条件复制并转发，所有主机都可以接收到所有信息，由于其不用路径选择，所以其网络成本可以很低廉。