## project 5 review 反馈

一、18、19号 DMA 寄存器与 3、4号 DMA 寄存器之间有什么的关系?

可以简单理解 18,19 号寄存器存的是当前描述符指针,3、4 号存的是描述符的头指针,18,19 号寄存器是只读寄存器,里面的值是硬件写入的。

以发送为例:我们把发送描述符的首地址放到 4 号寄存器里,当开始发包后, 当前发送描述符的值会被硬件写入 18 号寄存器中,每当完成一个发送包,当前 发送描述符的值会在 18 号寄存器里自动更新。

二、实验一、实验二、实验三分别收到多少个包打印?

实验一是收到多少个包就打印多少个包,打印完线程就退出;

实验二和实验三是收到64个包才打印。

框架里面的 check\_recv() 就是检查收到的数据包是否有错误,打印数据包的功能。

三、多个线程或进程在使用网卡的时候,网卡内 DMA 是串行的还是并行的?

我们提供的板卡上的网卡只有一个 DMA, 所以数据在 DMA 是串行的,可以理解为:哪个线程的接收/发送命令先到 DMA, DMA 就先搬移哪个数据。

四、对 1 、2 寄存器写任意值(推荐写 1)是连续写 64 次!!! 中途不用检查描述符的 own 位,相信我们的板卡,连续写 64 次是不会丢包的。

五、bonus 的难点

bonus 要在实验三的基础上实现,在高速的情况下怎么处理丢包等等。

六、clear interrupt 在哪使用,怎么操作?

clear interrupt 这个函数功能就是清除网卡的中断,在实验三中会用到。在网卡中断处理函数中需要调用 clear interrupt 这个函数,如果不调用就会重复进入网卡中断处理函数。

七、register\_irq\_handler()函数的功能是:注册网卡中断处理函数。

八、mac 结构体每个成员的含义如下:

```
typedef struct mac
```

```
uint32_t psize; // 每个数据包的大小uint32_t pnum; //数据包的个数uint32_t mac_addr; // MAC 寄存器的基地址uint32_t dma_addr; // DMA 寄存器的基地址
```

```
      uint32_t saddr;
      // 发送数据包的虚拟首地址

      uint32 t daddr;
      // 接收数据包的虚拟首地址
```

uint32\_t saddr\_phy; //发送数据包的物理首地址 uint32\_t daddr\_phy; //接收数据包的物理首地址

 uint32\_t td;
 // DMA 发送描述符的虚拟首地址

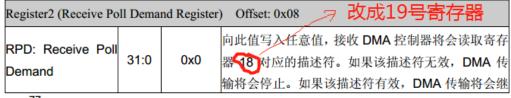
 uint32\_t rd;
 // DMA 接收描述符的虚拟首地址

uint32\_t td\_phy; // DMA 发送描述符的物理首地址 uint32 t rd phy; // DMA 接收描述符的物理首地址

## } mac\_t;

大家可以根据自己的设计自行修改 mac\_t 的成员变量、一些函数的入参等等。 九、龙芯 1c 的手册上有一个错误,在此更正一下。

11.1.3 Register2 (Receive Poll Demand Register)

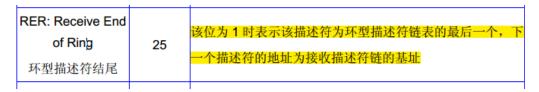


十、网卡中断需要多次使能吗?

不需要,只需要使能一次就行了。

十一、s\_reset(mac\_t \*mac)

这个函数的功能是复位 MAC 内部寄存器和逻辑,在网卡初始化的时候会用到。十二、RDES1 的第 25 位可以不置 1 吗?



## 可以。

十二、receive buffer 可以不清 0 吗?

## 可以。

十三、为什么用环形链表?

这样可以重复利用描述符,是目前的主流。

十四、在接收包的时候会先接收 6、7 个操作系统发的包,然后才能接收到 pktRxTx 发的包,请问这种情况需要怎么操作?

请丢掉操作系统发的包,只留 pktRxTx 发的 64 个包。描述符需要设置成 64 个,实现重复利用接收描述符。

十五、 $start\_code$  有一个地方写错了,在 mac.c 里面  $do\_net\_recv()$ 中,请大家修改一下。

```
uint32_t do_net_recv(uint32_t rd, uint32_t rd_phy, uint32_t daddr)

{

ptiDMA_BASE_ADDR

//PLEASE enable MAC-RX

reg_write_32(DMA_BASE_ADDR + 0x18, reg_read_32(GMAC_BASE_ADDR + 0x18) | 0x022000002); // sta
reg_write_32(DMA_BASE_ADDR + 0x1c, 0x10001 | (1 << 6));

//you should add some code to start recv and check recv packages

//you should add some code to start recv and check recv packages
```