（编号原则: 项目简称\_bug\_发现时间\_该项目当天发现的bug序号\_是否解决）

编号：xl-6004\_bug\_2014-7-14\_1\_ok

简介：与外部PPS对时，从网口EMAC0接收MU发来的报文，并打上时标，加上转发协议头部通过EMAC1转发。

发现测试：

1. 在接收SV报文时，加上抓取时标的配置之后，出现 EMAC FIFO OVERFLOW ；

查找原因：

1. 查阅bf609 硬件参考手册【1】，当接收数据达FIFO设置的阈值（64bytes）之后，数据通过DMA搬运到 用户内存buffer，只有当FIFO数据超过最大FIFO SIZE（RX:128BYTES, TX:256 BYTES）时，才会发生FIFO OVF。
2. 根据手册的描述，可以理解为：之所以发生FIFO OVF，是因为 FIFO数据超过64bytes 或到达帧数据结束时，FIFO里的数据没有搬运出去，结论：DMA搬运数据到用户内存buffer的操作不正常【2】。
3. 在EMAC中断响应代码中加入一些打印插件【3】，检查所有的状态位【4】，发现在打印“FIFO OVF”之前，首先打印一次 “no rx buffer”，断定是rx的buffer 不够，参考处理no TX buffer的处理（例程），并添加打印信息【5】，判断是buffer 不够还是 描述符不够【6】，从而决定增加相应的数量【7】。
4. 再次测试【8】，无FIFO OVERFLOW。

经验教训：

1. 理解系统；
2. 理解系统；
3. 不要想，而要看；
4. 检查插头；
5. 不要想， 而要看；
6. 理解系统；
7. 一次只修改一个地方；
8. 保持审计；

编号：xl-6004\_bug\_2014-10-25\_1\_ok

简介：与外部PPS对时，从网口EMAC0接收MU发来的报文，并打上时标，加上转发协议头部通过EMAC1转发。

发现测试：

1. 在接收SV报文时，通过PC从EMAC1 发送程序下载，609主板出现FIFO OVERFLOW ；

查找原因：

1. 查阅bf609

方法：

1. 在接收到下载文件的第一个数据包时，停止EMAC0（参考609 硬件参考手册进行，而不是直接disable EMAC），单板测试正常，但多板同时进行时，不一定成功。

教训：

1. 看手册。

编号：xl-6004\_bug\_2014-10-28\_1\_ok

简介：与外部PPS对时，对外部PPS的上升沿使用辅助快照打上硬件时标,当第一次上升沿到来时,重置609的系统时钟,PTP按照100M的参考频率进行计时。接下来，根据PPS的快照时间（609系统时间）与标准时间（认为PPS的上升沿是标准的1s开始）的纳秒部分差值，通过PI算法调整609系统时间，以使得609系统时间与外部PPS同步。

发现测试：

通过UART打印发现PPS的快照时间（609系统时间）与标准时间（认为PPS的上升沿是标准的1s开始）的纳秒部分差值在+/-10ns，但是609 输出的PPS与外部PPS的上升沿的差值不是这么多，而且会逐渐增大。（609 输出的PPS是609系统时间的物理表征，但是目前看到的现象并不正确）

Debug过程：

仔细阅读bf609的硬件参考手册【1】，发现代码对相关寄存器的配置都是对的，仿真器调试【2】 ，观察寄存器的值也都正确。

查阅Datasheet【3】， 发现“硬件辅助时间戳最高可提供12.5 ns的分辨率”，而我们设置的PTP参考频率是100M，即提供10ns的分辨率，超过了最高可提供12.5 ns的分辨率，将PTP参考频率该成50M之后，输出的PPS与外部PPS的上升沿的差值是正确的。

经验教训：

1. 理解系统；
2. 不要想，而要看；
3. 理解系统。