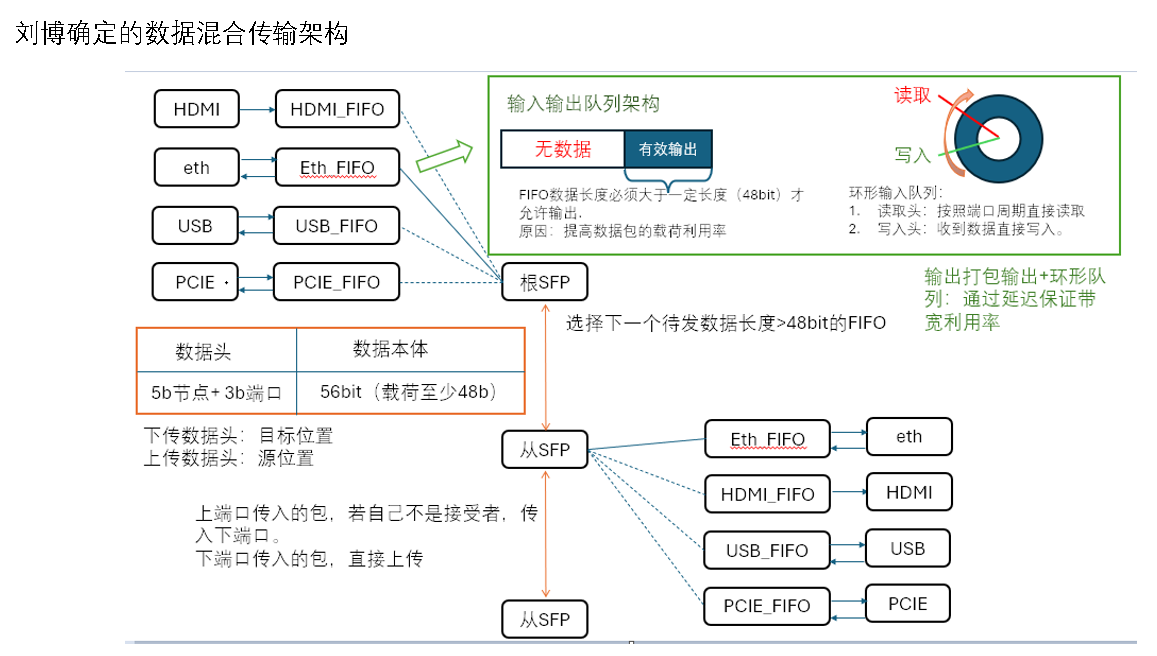
# 数据混合传输项目Hybrid Transport



2025.4.15，刘博确定数据混合传输架构。

1，前期先混合EHT+HDMI。

2，每天PUSH代码。

3，需要确定MVP。

4，每个代码文件规范，有功能说明。

4，子节点设备接入的时候，需要向根节点汇报。

6，前期确定输出验证时间，2025.5.16。

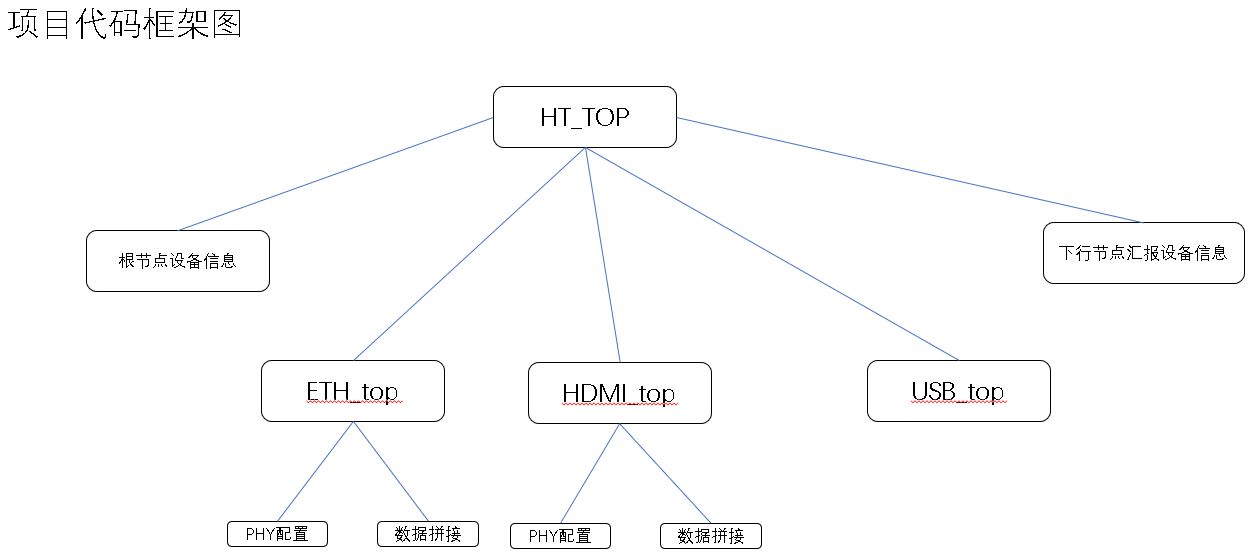
2025.4.16

1，项目代码总体框架。

2，运行原理。

3，分部实现，并验证。

4，后端需要足够FIFO深度，以防数据整包没有连续发送。



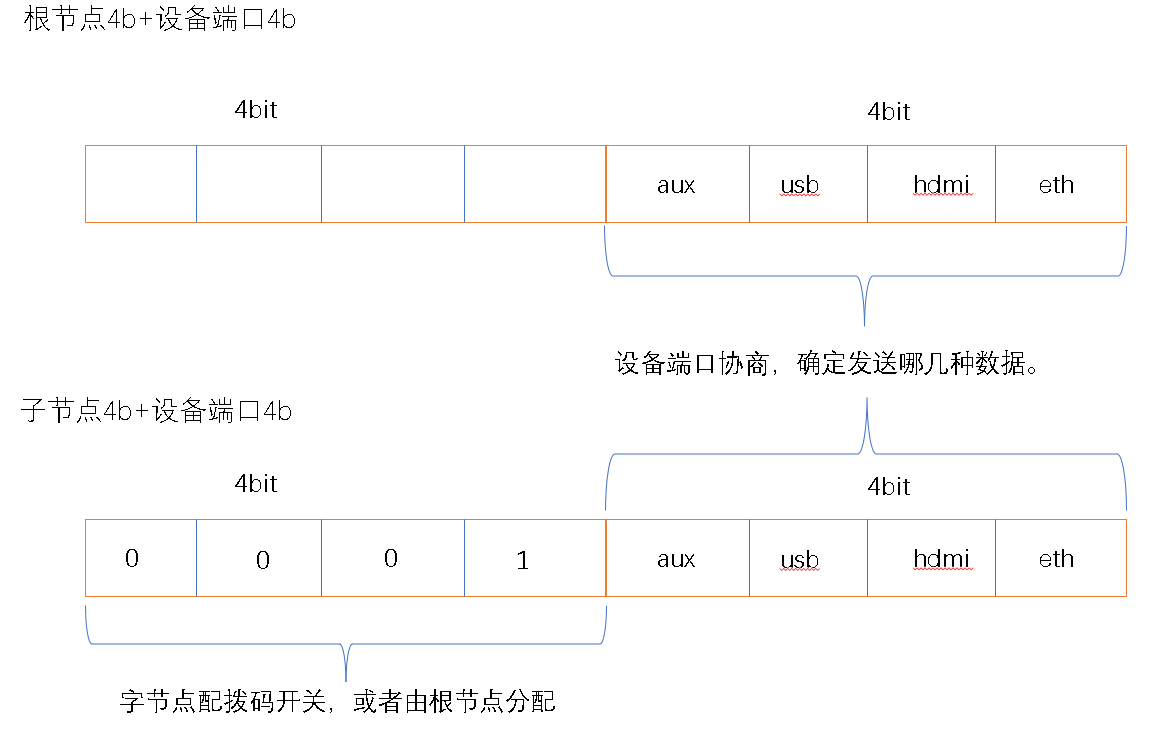
工作原理过程：

1. 根节点上电保存设备信息，并可以持续更新。
2. 保存下行节点设备信息。
3. 根节点根据节点信息+设备信息发送数据。
4. 子节点接收到信息后，节点信息对上就接收，没对上就转发下一节点。
5. 每个FIFO收集到的数据超过56bit,就提出发送申请。
6. 轮流发送每个FIFO数据，每发送56bit。
7. 当数据包结束的时候，不够56bit，在尾端补0。更新tkeep有效字节。（数据结束标志需要确认。）
8. 子节点FIFO数据需要等待一定数量才能读取发送，以免整包数据出现断续的情况。
9. 根节点和子节点辅助信息经过AUX进行交互。
10. 根节点需要一个SFP，子节点需要两个SFP。

ETH部分开始写代码，测试。（以前是用几个简单的缓冲器解决，数据位宽是64bit）

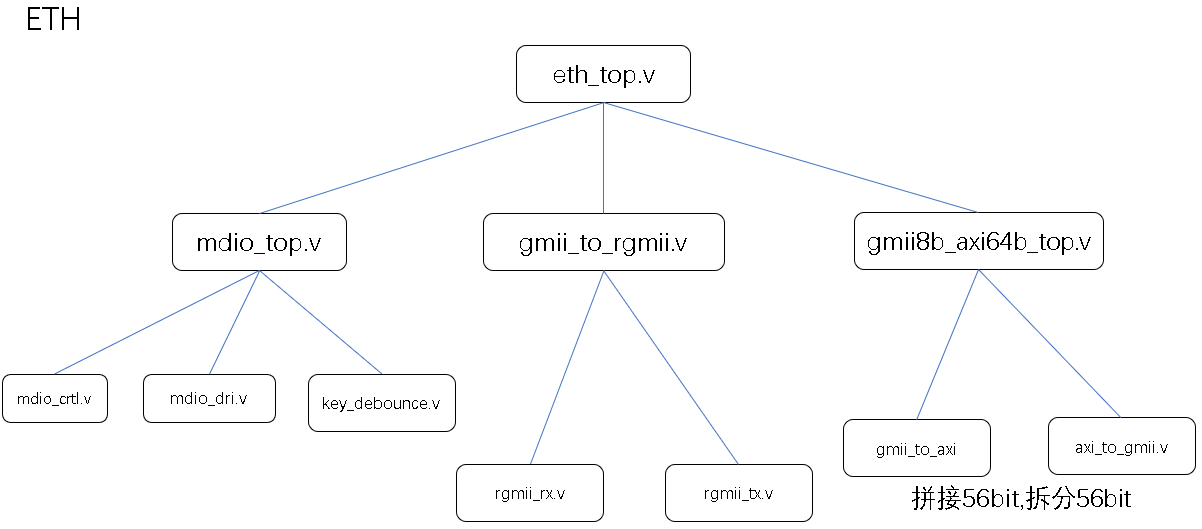
1, PHY配置，配置完成给出速率标志。待和子节点速率对比，速率不一样需要重新配置。

2，数据拼接，前8bit+56bit。前8bit:节点4b+端4b。



3，前4bit可以表示1-15个子节点，0根节点。一般节点数在10个以内。

4，ETH代码框架。



修改mdio\_top.v，要求：

1. 当本设备速率比子节点设备速率高的时候，按子节点的速率重新配置。
2. 当本设备速率比子节点设备速率低的时候，子节点的速率重新配置。

2025.4.17测试eth phy配置功能。暂时不管配置功能。

2025.4.22。编写总体代码框架HT\_TOP.v。

1, 实现功能，每种数据FIFO（eth\_FIFO,hdmi\_FIFO,usb\_FIFO,aux\_FIFO）到56bit,就提出发送申请。HT\_TOP按顺序轮流读取提出申请的FIFO数据给收发器IP核发送。

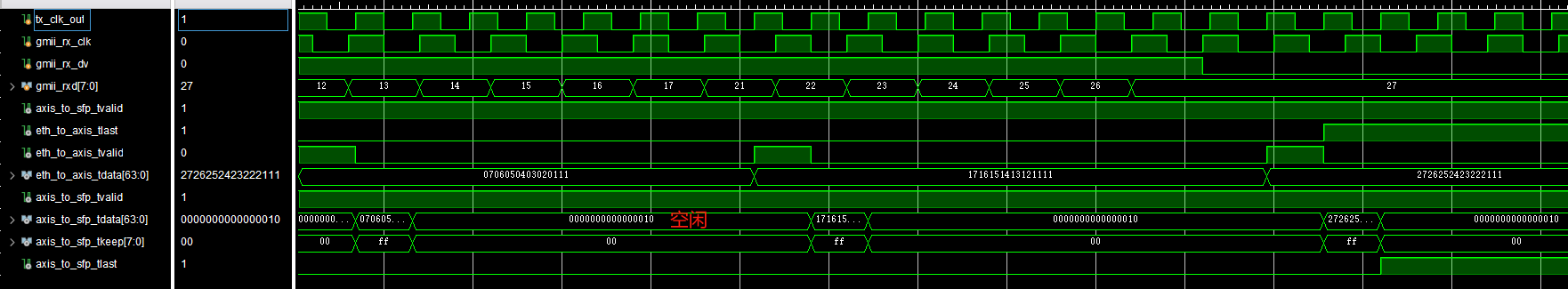
A,子节点默认发送aux信息，直到主机回复。

B,用16个8位reg数组接收各个子节点aux信息，包含子节点号，数据类型（eth,hdmi……）。

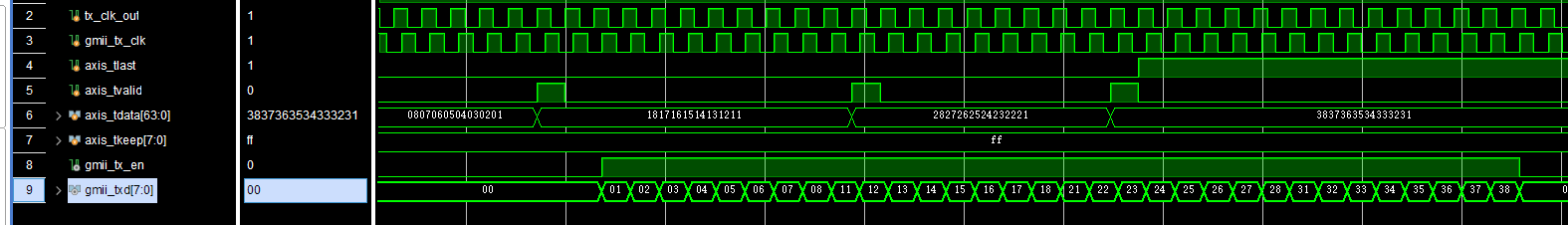
C,将有效的axu信息发到各类型数据处理模块。

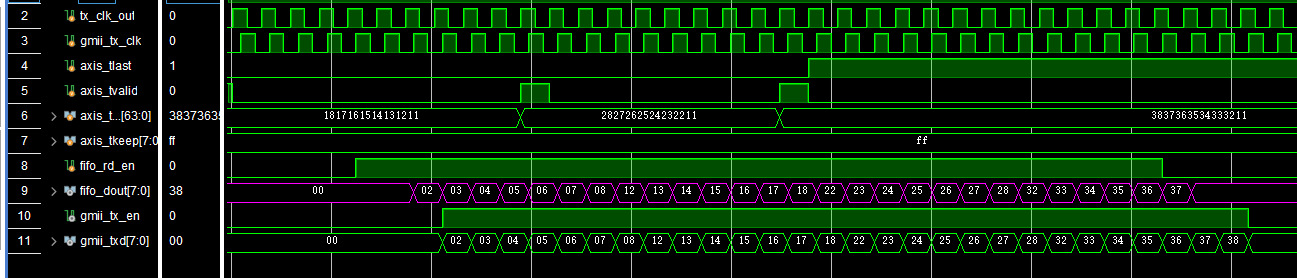
2，修改数据拆分和拼接。gmii8b\_axi64b\_top,数据由低8bit数据头+56bit数据组成。

A,测试gmii to axi程序。加入节点信息。仿真数据已经正确。



B,修改axi to gmii程序，使用FIFO。一个大坑，FIFO复位需要时间，仿真总是丢前几个字节，摸了半天才发现。





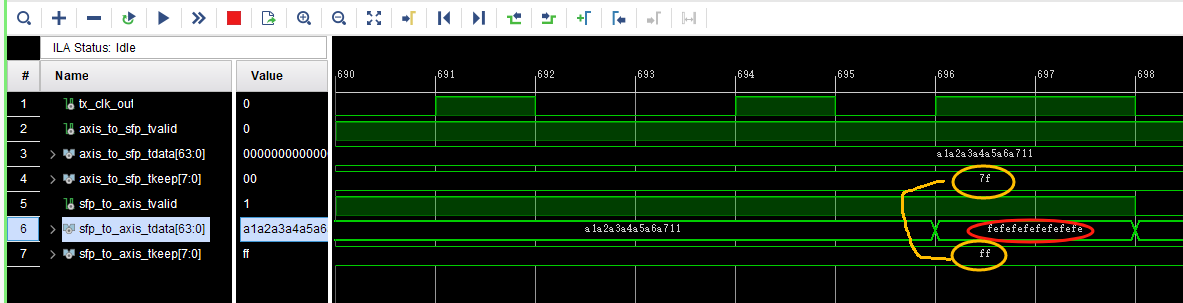
2025.4.29,发送和接收逻辑仿真完成，程序合并一起，编译。

1，发送端发现问题，每种类型数据FIFO有数据后，发出发送申请标志，数据在取走后，申请标志需要清除，不然一直会有发送申请。

1. FIFO写入字节数，最高位是Z，需要去掉。

2025.5.6

1,接收端没有接收到任何数据，检查收发器IP核配置.



测试发现，数据包结束时候IP核自动加入fefefefe fefefefe. 并且tkeep在tlast无效时候，不能指示有效字节，全是FF。

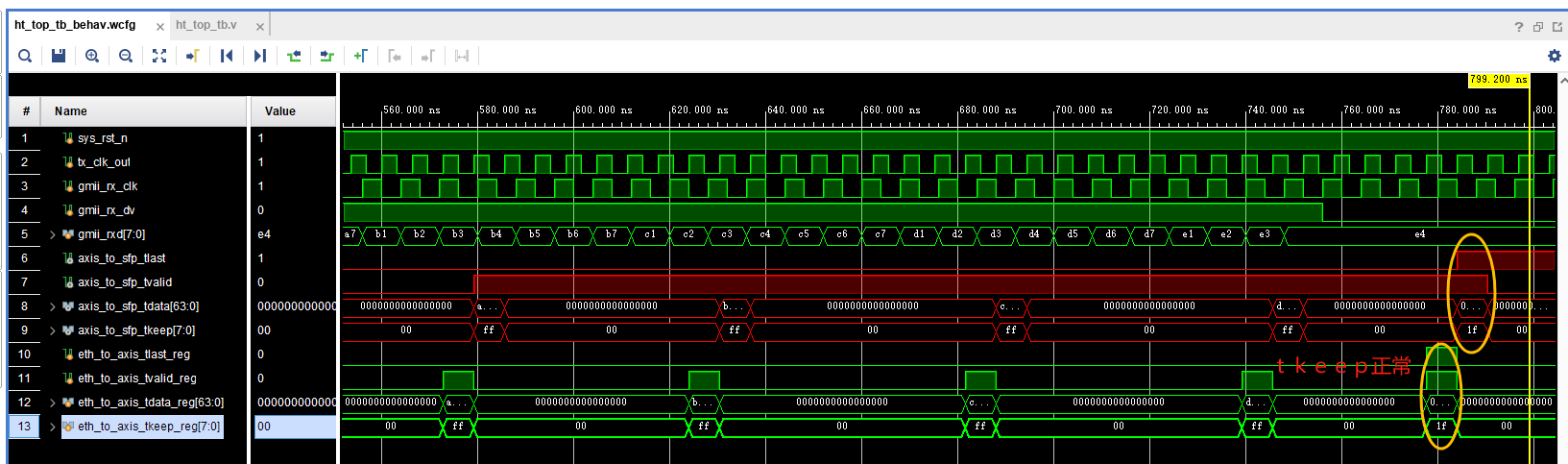
IP核自动加入fefefefe fefefefe与tlast和tvalid有关。

当tlast无效，tvalid为0时IP核自动加入FE，并把Tvalid延迟一个周再设置为0。

当tlast有效时候，IP核不会加入fefefefe fefefefe.

Tkeep只在tvalid无效0，tlast有效1的时候可以正确指示有效字节，用来表示最后一个8字节中哪几个字节有效。IP核决定的。

仿真数据。



上级测试发现tvalid和tlast不正常。经过层层查找，终于找到问题，tlast接口没对上。修改后数据正确！

