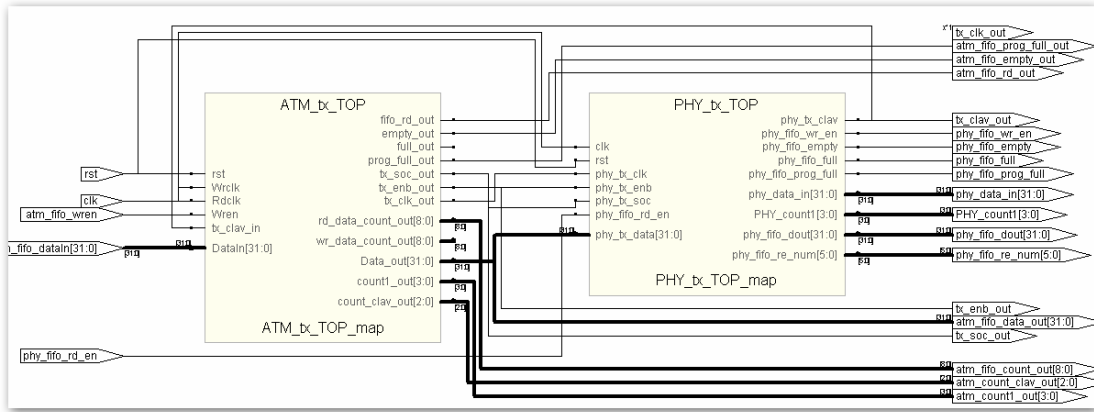


UTOPIA L1 的 ATM--->PHY 也就是标准 utopia1 的图形方向



Utopia 在芯片内部的仿真，一个 ATM 层的，包括一个异步 fifo 和 UL1 的 ATM_tx 接口，如图 2。

一个是 PHY 层，包括一个异步 FIFO 和 UL1 的 PHY_tx 层的接口。

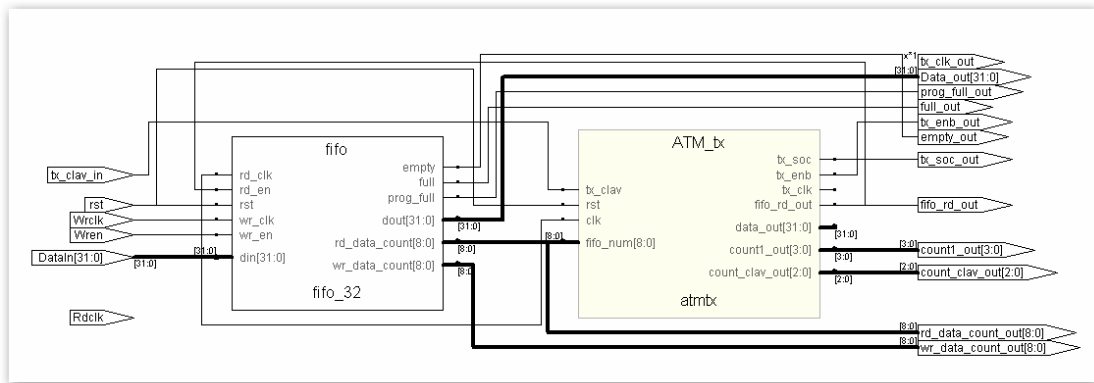


图 2: ATM 侧

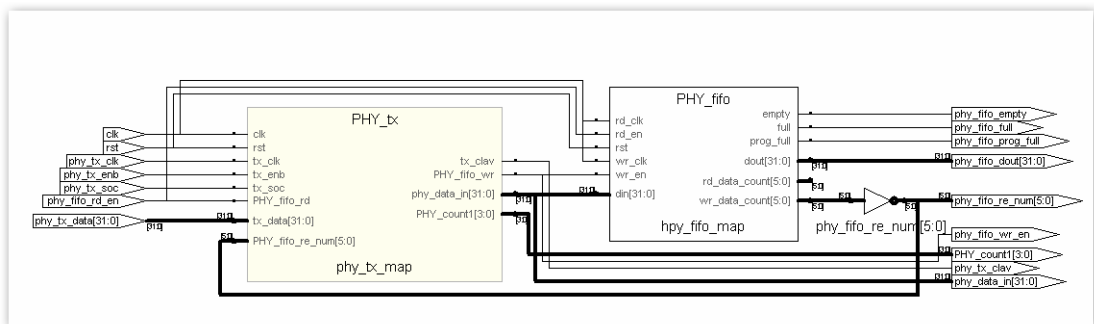


图 3: PHY 侧

一. UL1 的 TX 的信号:

Txdata: (略去)

Txsoc: 由 ATM 层给出, 表示 txdata 中的数据为信元的第一个有效字节。

Txenb: 当 txdata 中数据有效时, 由 ATM 层给出。

Txfull/txclav: 对于字节模式: 当 PHY 层可以接收多于四个字节的数, PHY 层将会置错了 txfull 为高。

对于信元模式: 当 PHY 层有多于一整个信元的空间, PHY 层将会置错了 txclav 为高。

TXclk: 略

二. 操作和时序

发送接口是由 ATM 控制的 (clk 由 ATM 层提供), 这就要求 PHY 层的 fifo 必须和 ATM 的比特率配合。

工作流程如下: 1.PHY 层通过置 txclav 为高, 指示能够接收一个完整的信元(四个字节), 2.然后 ATM 层会检测这个 txclav, 并根据内部的 fifo 的空间情况 (就是判断有没有一个完整的信元), 如有, 则会发送信元, 通过置 txsoc 和 txenb (当然提前要先读出 atm_fifo 中的数据。这也就是 txclav 为什么要在一个信元的最后置低。在最后的第三个字节处判断是否可以再次发送信元, 然后 ATM 层在最后第二个得到 clav 为高, 知道 PHY 有能力接收下一个完整的信元, 此时置 atm_fifo 的读有效, 读有效会在上一个信元的最后时钟有效, 读出的数据会在新的一个信元的第一个时隙有效。)

2.1 字节的握手协议:

在一个“时间周期”, 也就是“发送窗口”, PHY 层会存储 txdata, 当 txenb 有效的时候。(暗示: 只要 txenb 有效, 那么 PHY 层就会认为是有效的。所以 ATM 层在给出 txenb 是必须保证数据是有效的, 也就是 txenb 不能随便置为有效)。

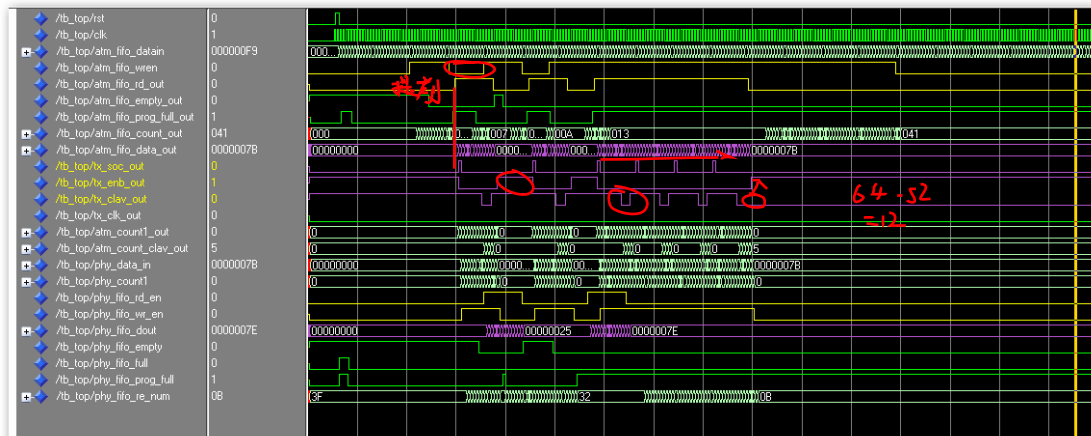
发送窗口的含义: 从 txclav 被置为有效后 (表示它能够接收数据, 不一定是完整的信元, 只要大于四个 byte 就 ok), 到 txclav 被置为无效后的“四个有效的写时钟”(写入四个有效的数据)。

Phy 能够在任何时刻置 txfull 为低 (无效), 一旦置低就表示 ATM 层不能传送多于四个 byte 的数据 (可以是四个)。

在置低 txclav (变无效) 以后的四个写有效的时间内, ATM 层必须置高 txenb (无效)。并且 txenb 在 txclav 被检测重新置高 (有效) 前, 不能置低 (有效)。在“发送窗口”之外的 txenb 有效被认为是“错误”, 并忽略这样的写入的数据。在发送窗口内, ATM 层才能对 txenb 有效或无效。

2.2 信元的握手协议

和字节的握手协议全部一样, 除了一项例外。就是一旦 txclav 被置有效 (高), 那么就表示 PHY 层有能力接收一个完整的信元, (也就是 PHY 层只有在确认能完整接收一个完整的信元时才能置 txclav 有效。) txenb 能够背 ATM 层用来作为字节级的流量控制, (就和字节的握手协议一样) ? ? ? ?。(表示 ATM 不一定连续发送一个完整的信元?? 可能允许中间有间隔)。为了确保 ATM 层不会“过量”传输 (就是此时 PHY 层已经没有空间), PHY 层必须在 4 个有效的写时钟前置 txclav 无效 (见标准的图示, 多是这样的), 以保证背靠背的传输。(就是“连续“不间断的发送信元)



测试方法: 1: ATM 端两个信元间隔任意个时钟输入, 擦看个 UL1 的信号是否正确, 能否

连续传送，结论可以的。

2: 连续多个信元到达，能否连续发送出去，查看个信号是否正确。结论正确。

3: PHY 侧的 fifo 满了之后，clav，enb，soc 的反应是不是正确？正确的。

4: ATM 层 fifo 空时的反应？是否正确。正确的。如下图（enb 根据一定的条件会变高）

