**DMA数据分割器**

|  |  |  |  |
| --- | --- | --- | --- |
| Document Version Control | | | |
| Version | Date | Author | Changes |
| 1.0.0 | 2023/8/25 | 王雨霄 | 实现了DMA数据分割器 |
| 1.0.1 | 2023/10/21 | 王雨霄 | 增加AXI-DMA配置教程 |

1. **功能概述**

DMA进行S2MM流向的数据传输时，必须由AXIS输入端口中TLAST信号的拉高来标志本次数据传输的结束；若通过逐级传递的方式将TLAST信号传入DMA中，则PS端无法主动控制单次DMA读取的目标数据量，为解决这一问题，本设计对DMA数据分割器进行了HLS实现，主要功能特性如下：

1. 支持AXI-Lite控制接口；
2. 支持AXIS数据接口；
3. 可为任意长度数据流的末端数据添加TLAST信号；
4. **接口概述**

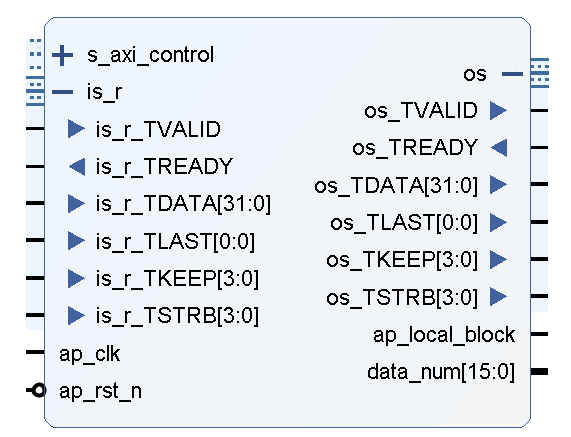
****

图1. 模块接口



图2. 接口功能表

注：

若前级模块的AXIS输出接口仅包含“xxx\_tdata”、“xxx\_tvalid”及“xxx\_tready”三个基本信号，则本模块AXIS输入接口中的“is\_r\_TKEEP”及“is\_r\_TSTRB”信号需接为全1（4‘b1111），“is\_r\_TLAST”信号可悬空或接为0。

1. **使用方法**

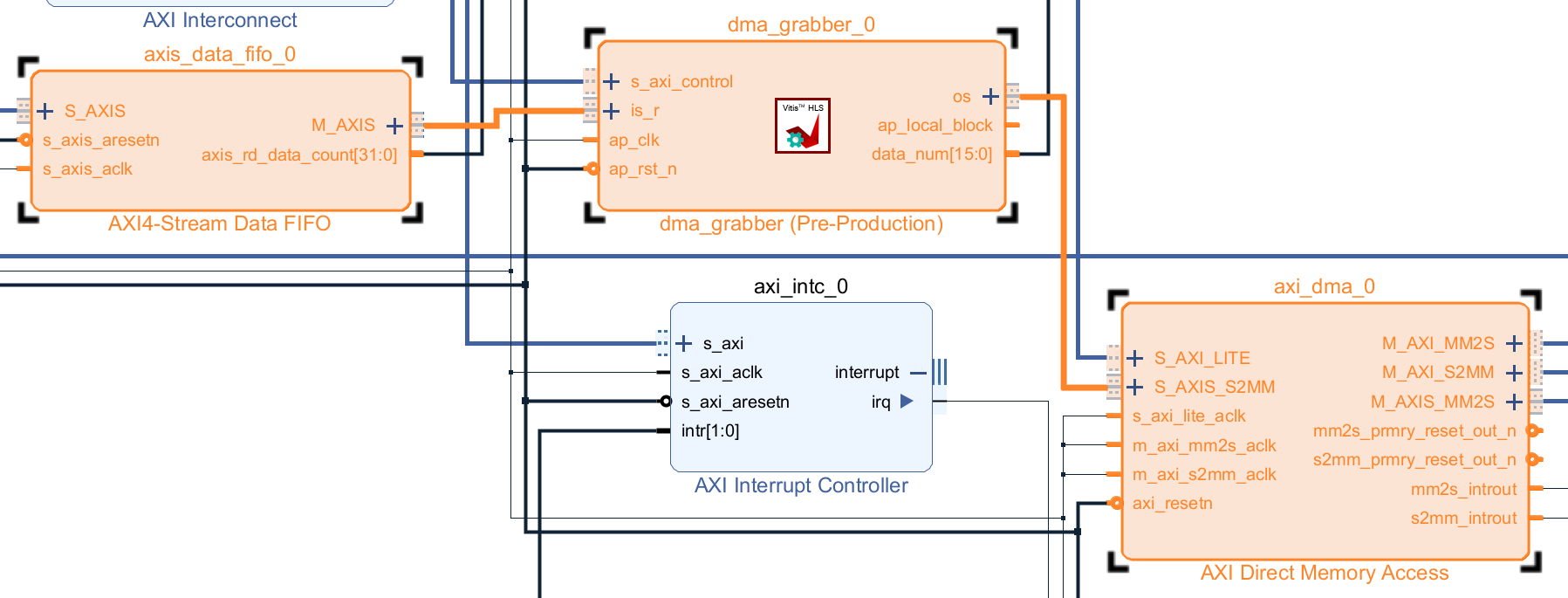
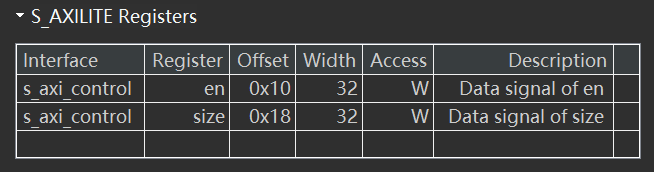


图3. 模块连接图

dma\_grabber模块的接口可分为数据接口、控制接口、状态接口三种——其数据接口is\_r及os应位于“fifo-dma\_grabber-DMA”通路中，控制接口s\_axi\_control可作为从机通过AXI Interconnect挂载至总线，状态接口输出的计数信号data\_num可通过AXI GPIO挂载至总线（将PL端本次传输至DMA的有效数据个数告知PS端，实现软硬件状态同步）。

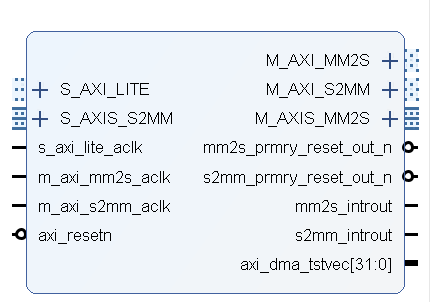
s\_axi\_control控制接口中包含如下寄存器：



在使用过程中，按照“en=0、size=目标数据量、en=1”的次序进行一轮寄存器写入，即可启动dma\_grabber模块的数据传输。

**附录：AXI-DMA配置**

AXI-DMA核的接口如下图所示：

****

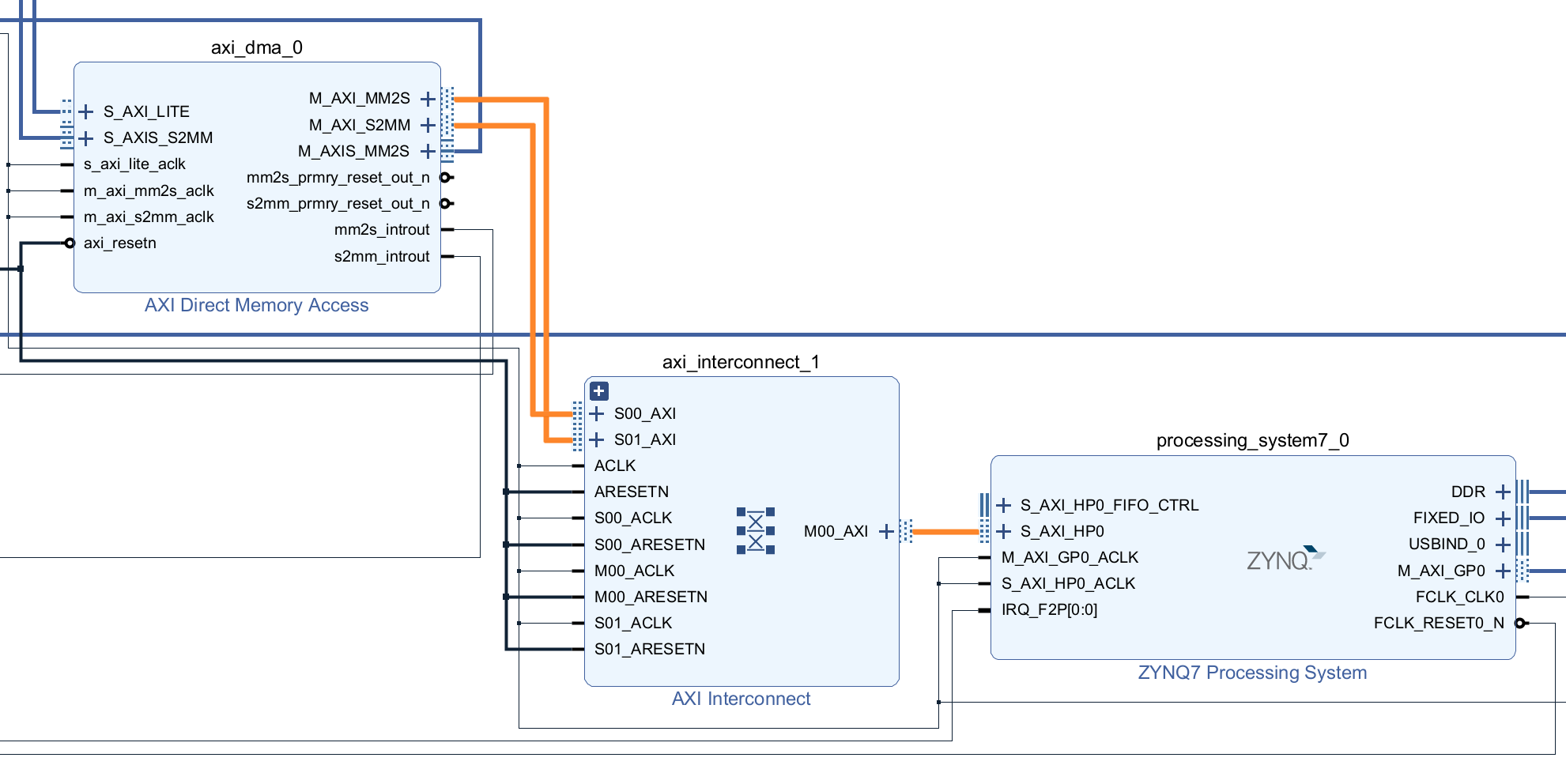
**PL**

**PS**

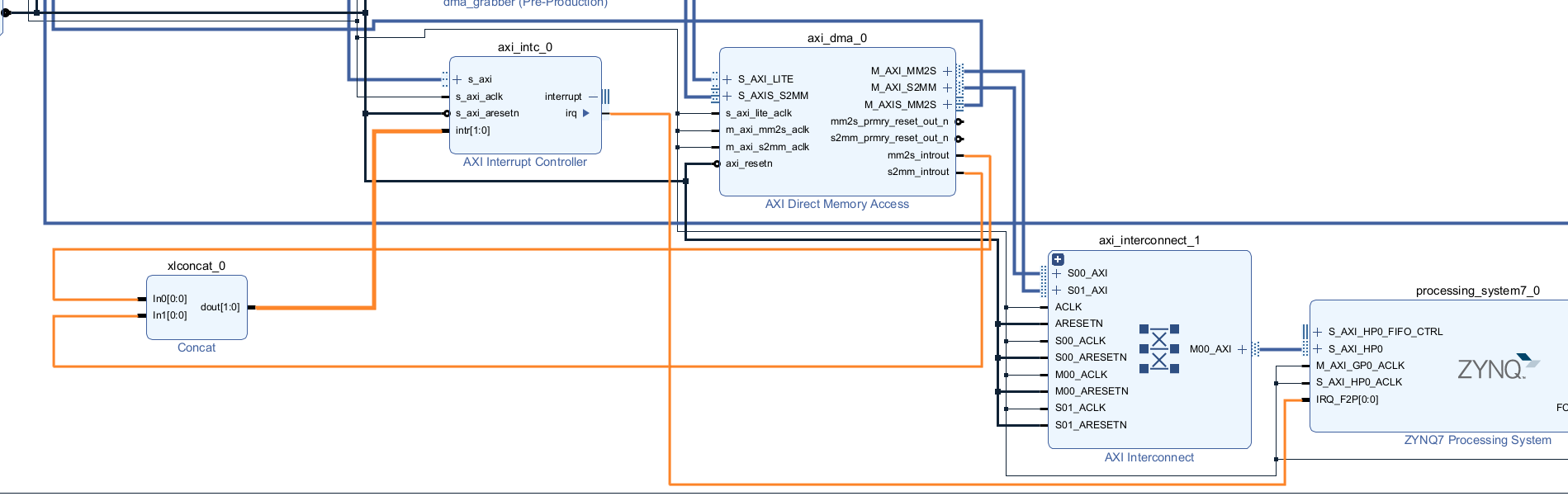
**PL**

**PS**

其中，“S\_AXIS\_S2MM”-“DMA”-“M\_AXI\_S2MM”通路用于PL端向PS端传递数据，“M\_AXI\_MM2S”-“DMA”-“S\_AXIS\_ MM2S” 通路用于PS端向PL端传递数据（ “S2MM”中，“S”意为“Stream”，指的是PL端数据流；“2”意为“to”；“MM”意为“Memory Map”，指的是PS端存储空间）；“M\_AXI\_S2MM”和“M\_AXI\_MM2S”应连接至ZYNQ核的**HP**接口上：

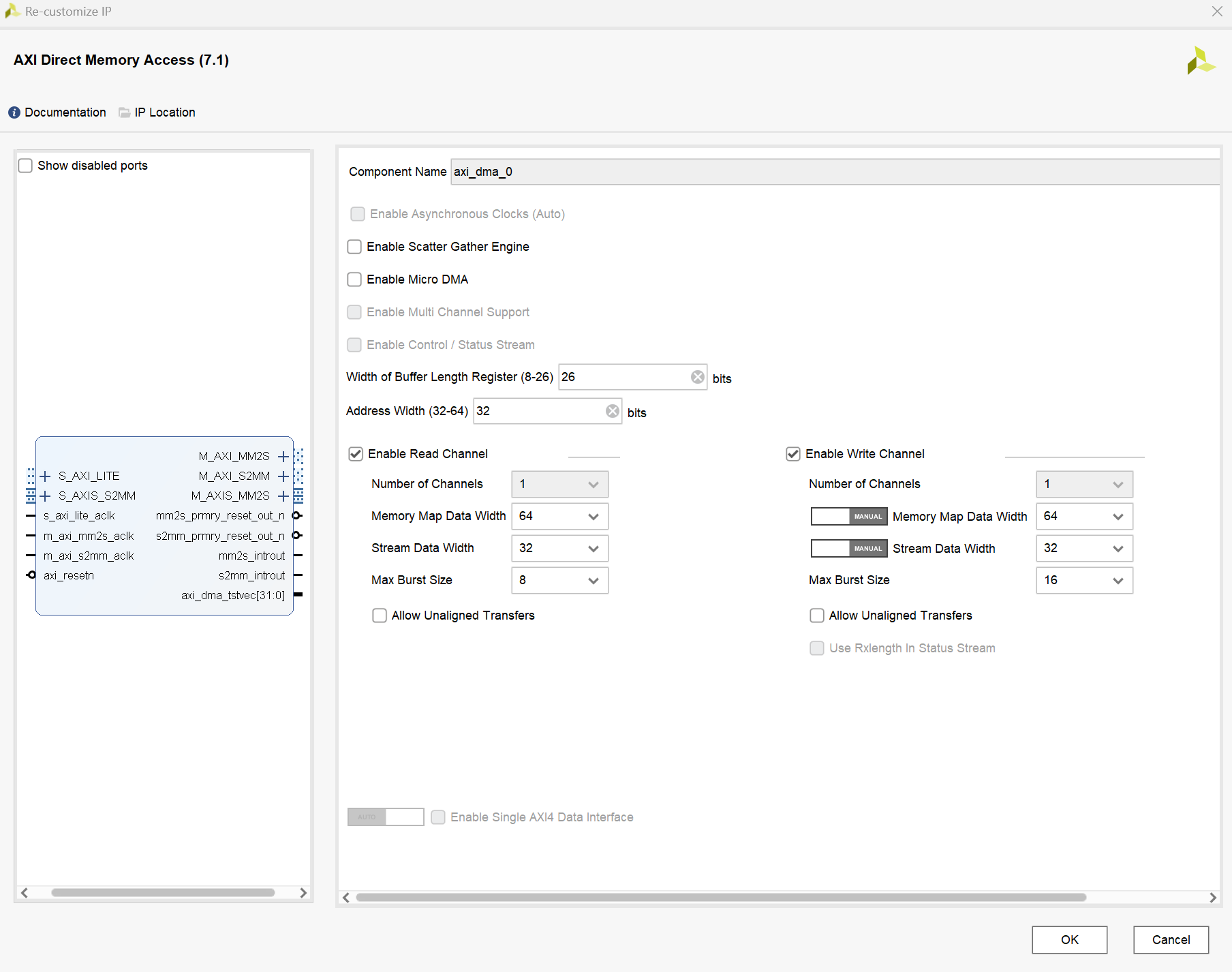


“mm2s\_introut”和“s2mm\_introut”用于输出中断信号，这两个中断信号必须按照如下通路，通过INTC核连接至ZYNQ核的中断口上：

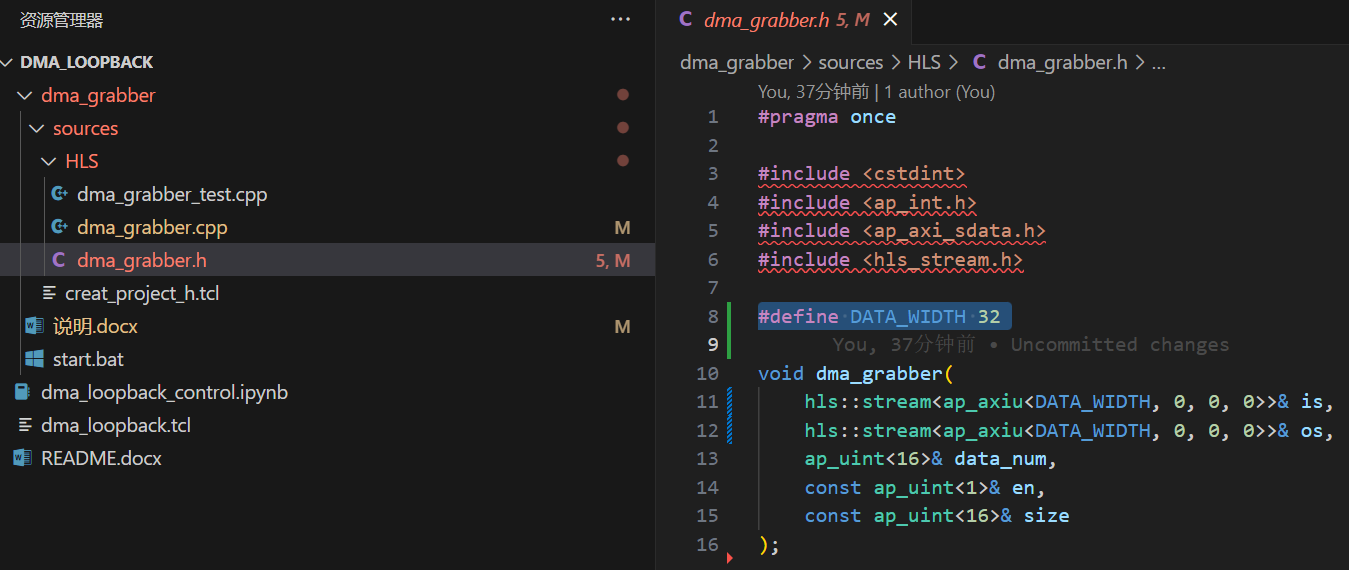
****

注：AXI-DMA的S\_AXI\_LITE控制接口与INTC的s\_axi控制接口都应挂载至ZYNQ的GP主机口上。

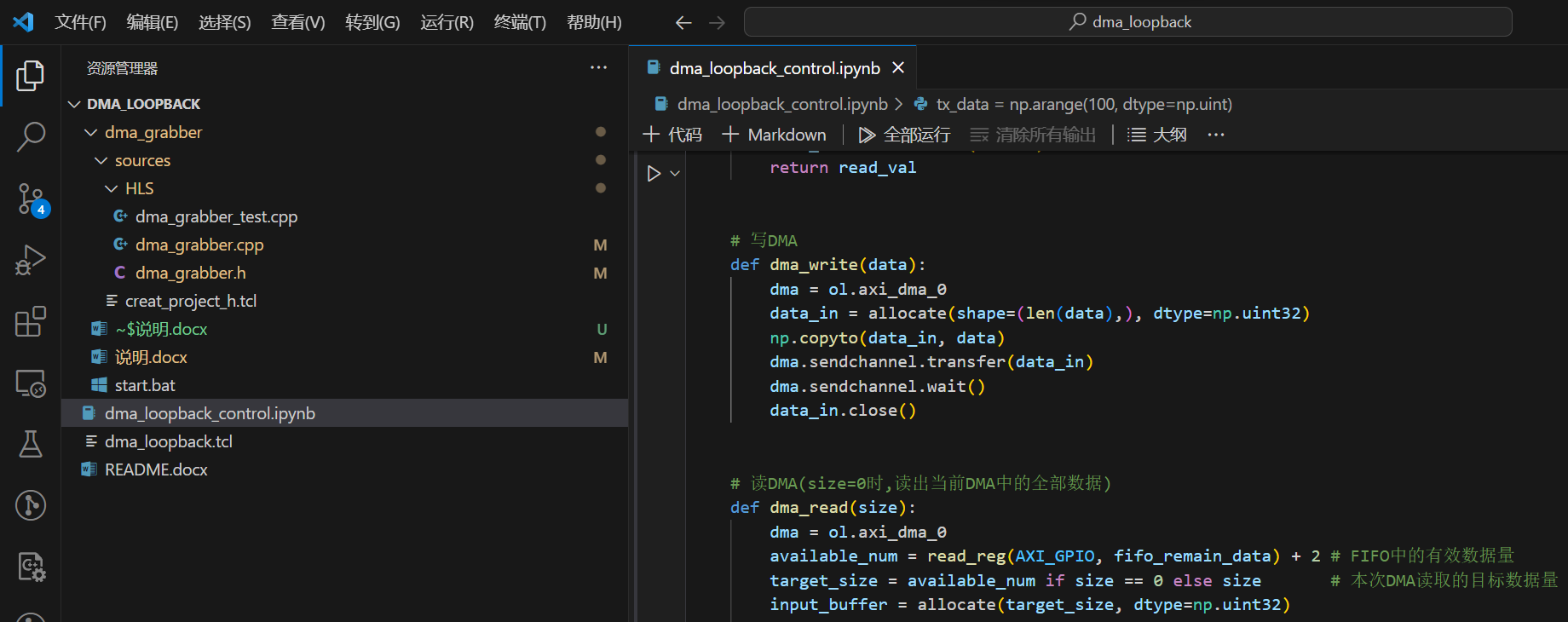
AXI-DMA核的配置参数如下图所示：



当使用PYNQ作为PS端时，红框中的PS端数据位宽必须设置为64；在此前提下，蓝框中的PL端数据位宽可设置为32或64，在示例工程中该参数设置为32；若需支持64位的PL端数据位宽，则需对dma\_grabber.h源码及python驱动代码进行如下修改：



**将32修改为64**



**将两处np.uint32修改为np.uint64**