使用PCIE更新ZYNQ的QSPI Flash参考设计

作者: Jackie Gao

# 简介

ZYNQ-7000的S\_AXI端口提供了外设访问PS内部外设控制器的接口，这其中包括4个S\_AXI\_HP端口以及两个S\_AXI\_GP端口。一般来说，可以访问的内部资源包括DDR控制器，QSPI控制器，OCM，IIC，SPI等。

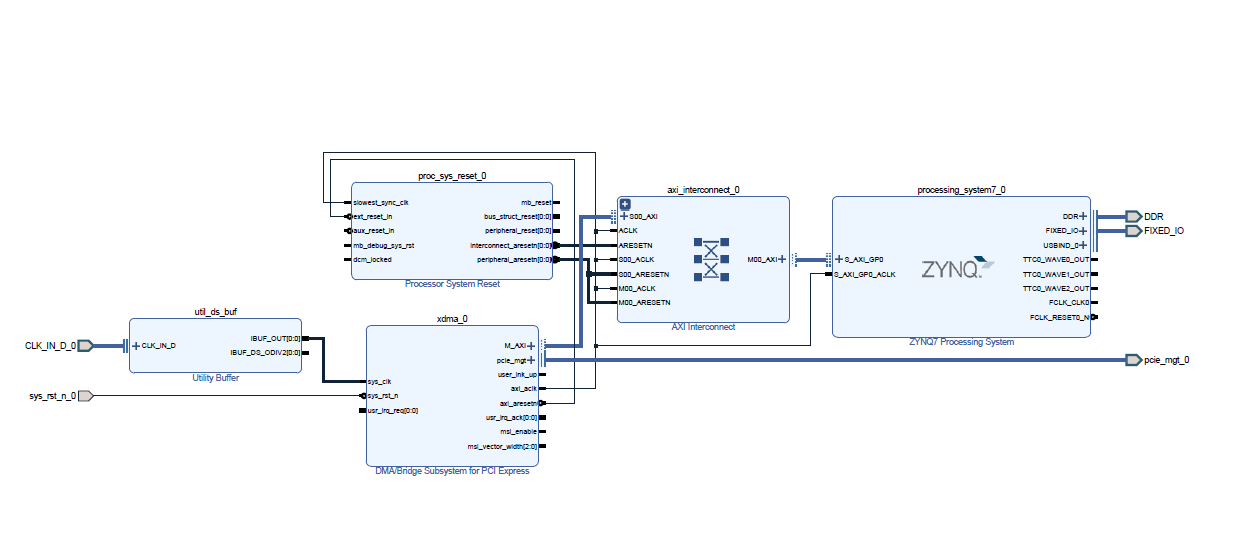
本教程提供一个最小的参考设计，使上柆机可以通过PCIE端口，通过S\_AXI\_GP0端口访问ZYNQ内部的QSPI控制器，用来更新Flash的启动分区。

软件方面，移植了embeddedsw的相关驱动代码至xdma平台，并提供了一个简单的上位机测试程序，用来烧写启动镜像至Flash分区。

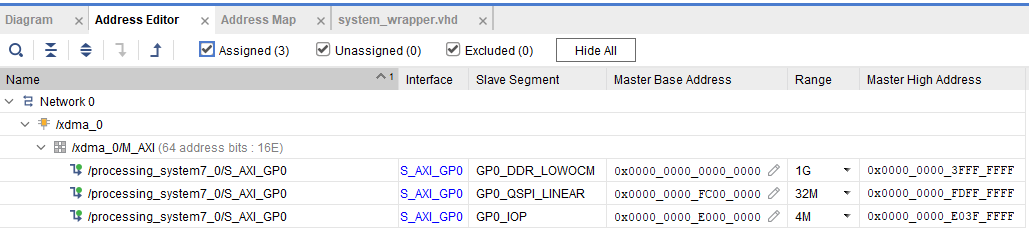
教程中所有涉及的设计及代码均在Windows和Linux（Ubuntu20.04）平台下作了验证。

# 系统设计

1. 系统设计框图参考如下。在设计实例化了一个XDMA IP，通过其M\_AXI端口连接至ZYNQ的S\_AXI\_GP0，以访问PS QSPI控制器。

****

1. BD中需要分配XDMA的M\_AXI地址如下，可以用Vivado的自动地址分配工具完成。

****

1. 如果有访问其它外设的需求，详细的地址可以参考UG585的Table 4‐1: System-Level Address Map

# 环境需求

## 硬件环境

* ZC706开发板
* Windows10/11或者Ubuntu20.04主机

## 软件环境

* Vivado Design Suite 2022.2
* Petalinux 2022.2
* CMake > 2.15
* Windows + Visual Studio或者Linux + GCC

# 代码结构

## 硬件

* hw文件夹里面里面包含vivado工程，直接用vivado 2022.2打开

## 固件

* 此参考设计用petalinux制作系统启动镜像，完整的petalinux工程位于fw目录下。
* 执行petalinux-build可以编译
* 执行petalinux-package –boot –u-boot –force –fpga XXX.bit 生成BOOT.BIN文件

## 软件

* qspi 的驱动软件移植自[embeddedsw](https://github.com/Xilinx/embeddedsw.git)，主要的改动包括修改xil\_io.c文件，用xdma的驱动替换默认的Xil\_In32和Xil\_Out32的实现。
* 软件支持windows和Linux的主机，采用cmake环境。如果在windows下，可以选择Visual Studio作为编译工具。编译过程如下:

1. cd sw
2. mkdir build
3. cmake .. && make

# 测试结果

* 在petalinux下生成的BOOT.BIN文件放在生成的上位机软件目录下，执行后可以看到更新flash的进度条在变化，说明FLASH被正常少烧写。
* 完成后切换到QSPI FLASH启动模式，上电观察启动过程。