# PCIe DMA IP仿真验证环境说明

PCIe DMA IP的设计代码来源于github上的一个开源项目verilog-pcie，网址为：<https://github.com/alexforencich/verilog-pcie>。在IP的设计过程中，继承了verilog-pcie的部分代码，并增加了功能，重新封装了IP接口。在仿真部分，完全继承了其原有的验证体系，整体上采用Python代码主控仿真过程，新增了仿真激励部分数据生成模块（verilog代码编写）。

## 仿真环境的安装

Python和verilog代码进行联合仿真，需要用到MyHDL的开源库。Verilog的仿真器使用开源的Icarus Verilog软件。仿真需要软件的安装过程：操作系统以Ubuntu-18.04为例

1. Python需要安装Python3，执行sudo install python3
2. 安装pip软件，执行sudo install python3-pip
3. 安装myhdl库，执行pip3 install myhdl
4. 安装iverilog软件，执行 sudo install iverilog
5. 安装iverilog软件的myhdl.vpi文件，下载myhdl源代码（http://www.myhdl.org），在其/cosimulation/icarus目录下执行make，得到myhdl.vpi文件，将其copy到/usr/lib/x86\_64-linux-gnu/ivl目录下
6. 在verilog-pcie/tb目录下执行python3 pcie.py，如果不报错，则环境安装完成

DUT设计代码如下，其中顶层为pcie\_dma\_c2h\_top模块。

./c2h\_dma/arbiter.v

./c2h\_dma/priority\_encoder.v

./c2h\_dma/dma\_if\_pcie\_us\_wr.v

./c2h\_dma/dma\_psdpram.v

./c2h\_dma/dma\_client\_port.v

./c2h\_dma/pcie\_dma\_c2h\_top.v

仿真代码如下，其中顶层为test\_dma\_c2h.py文件。

./c2h\_dma/gen\_frame\_sim.v

./tb/test\_dma\_c2h\_tb.v

./tb/fifo\_read\_monitor.v

./tb/sim\_define.vh

./tb/test\_dma\_c2h.py

./tb/pcie.py

./tb/pcie\_usp.py

./tb/dma\_ram.py

./tb/axis\_ep.py

仿真运行的命令为python3 test\_dma\_c2h.py，如果结果正确，则打印“diff pass！”

## 仿真平台的构成

在Python层，test\_dma\_c2h.py文件的bench函数中定义了整个仿真环境的顶层，包括pcie的rc设备及其存储空间，pcie\_usp的PCIe层处理设备dev，并创建端口，与DUT设备连接。DUT设备是用iverilog联合仿真方法生成的设备，代表着test\_dma\_c2h\_tb.v下面的verilog代码实现的逻辑功能。Bench函数内定义了时钟/复位等激励信号，check函数中定义了仿真操作的过程：rc设备的枚举过程，第一次PCIe写的仿真过程，保存rc接收到的数据，第二次PCIe写的仿真过程，保存rc接收到的数据，进行结果文件的对比，如果不一致则报错。程序运行的接口从“\_\_main\_\_”开始，调用test\_bench执行仿真，调用diff命令进行文件比较。copy\_mem\_to\_file函数从指定的pcie\_addr开始，将rc缓存空间中收到数据帧保存到文件中。该函数连续扫描整个缓存空间，遇到空缓存区则停止。默认发送数据的前两个字节为帧长，帧长等于0或大于4096则认为缓存空。

test\_dma\_c2h\_tb模块中包含了pcie\_dma\_c2h\_top模块作为真正的被测实体，以及gen\_frame\_sim模块产生所需的激励信号。Tb模块中和python层之间的信号交互通过$from\_myhdl和$to\_myhdl函数完成。fifo\_read\_monitor模块监控read\_frame\_\*信号接口，将发送给DUT的数据帧保存到文件中，按照一行保存一帧的原则，便于和rc端的保存文件进行比较。gen\_frame\_sim模块支持多种数据帧的生成模式，通过参数端口的配置，实现多种仿真模式的配置。

|  |  |  |
| --- | --- | --- |
| 参数名称 | 选择范围 | 说明 |
| DEFAULT\_FRAME\_LEN | 48-4096字节 | 默认帧长，当FRAME\_LEN\_TEST\_MODE选择=FIX时，使用此参数设定发送帧长 |
| DEFAULT\_TAG\_VALUE | 0-255 | 默认TAG值 |
| DEFAULT\_QUEUE\_DEPTH | 大于等于0 | FIFO中保存的数据帧深度，发送完成，FIFO即为空，每次复位后，FIFO中数据深度恢复此值 |
| FRAME\_LEN\_TEST\_MODE | 0-3 | "0=INC","1=DEC","2=FIX","3=RAND"，递增/递减/固定/随机 |
| FRAME\_CON\_TEST\_MODE | 0,1,3 | "0=INC","1=DEC", "3=RAND"递增/递减/随机 |
| FRAME\_LEN\_MIN | 48-4096字节 | 当FRAME\_LEN\_TEST\_MODE选择!=FIX时，使用此值作为发送帧长的最小值 |
| FRAME\_LEN\_MAX | 48-4096字节 | 当FRAME\_LEN\_TEST\_MODE选择!=FIX时，使用此值作为发送帧长的最大值 |

在./tb/sim\_define.vh中，定义了多种测试组合，可以在此文件的头部定义`define TEST\_xxx选择仿真用例，也可以创建新的仿真用例。