Скрипт Net2axis

Автор: **0xBADC0FFE**

Рецензент: **KeisN13**

2019 г.

Оглавление

Аннотация

При разработке устройств на FPGA часто возникает необходимость симуляции сетевого трафика. Избавиться от столь рутинной процедуры вам поможет скрипт Net2axis.

Описание работы

Данный скрипт позволяет генерировать AXI-Stream пакеты из файлов *.pcap (packet capture). Файлы pcap можно получить захватив сетевой трафик например с помощью Wireshark/tcpdump либо воспользовавшись конструктором пакетов PackEth.

Для запуска Net2axis необходим Python 2.7.X и Scapy 2.X. Работа скрипта сводится к двум этапам:

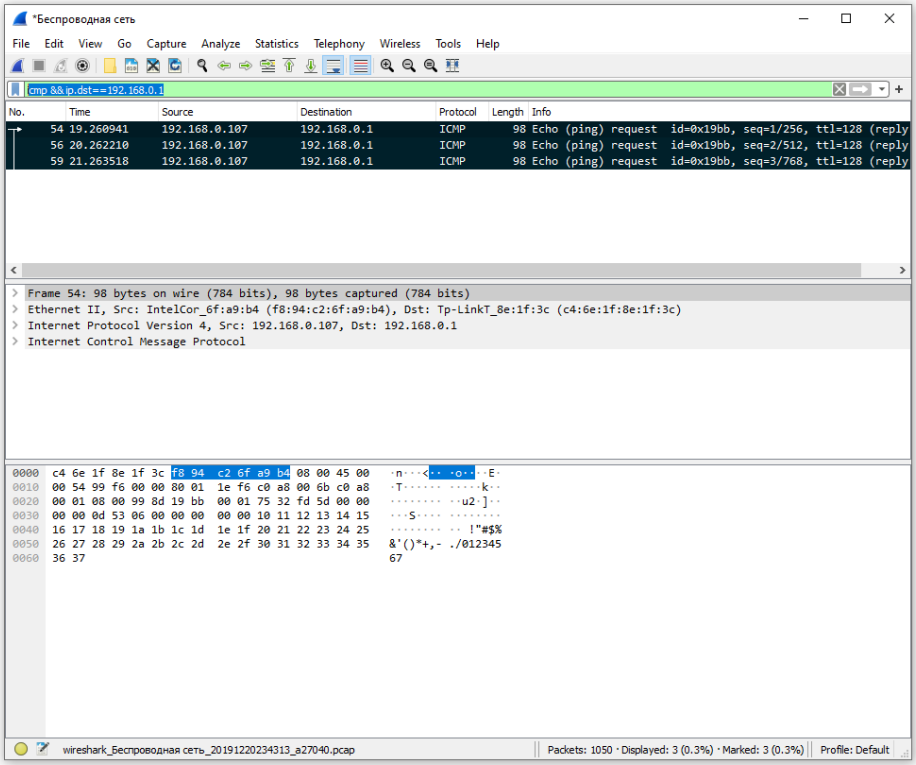
1. Файл *.pcap при помощи net2axis.py конвертируется в *.dat;
2. Полученный файл подключается к AXI-Stream генератору net2axis_master.v.

Демонстрация работы

Чтобы продемонстрировать работу скрипта захватим несколько пакетов ping (протокол ICMP). Запускаем Wireshark и выбираем активный сетевой интерфейс в меню Capture -> Options, затем жмем Start.

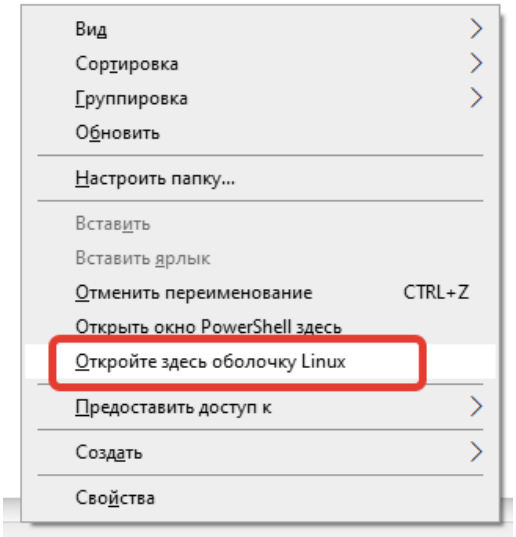
Во время процедуры захвата пакетов я отправил несколько ping-запросов к моему роутеру, после чего остановил Wireshark выполнив команду Capture → Stop.

Чтобы из всех захваченных пакетов отобразить только интересующие нас ICMP я задал фильтр следующего содержания: `icmp && ip.dst==192.168.0.1`



Экспортируем отфильтрованные пакеты в файл ping.pcap, для этого в меню File выбираем Export Specified Packets.

Теперь можно запустить скрипт Net2axis. Сейчас в моем распоряжении ноутбук с Windows 10 и установка scapy предвещает танцы с бубном, поэтому я воспользовался эмулятором Ubuntu 18.04 из Microsoft Store. Сразу после установки эмулятора открываем проводник, переходим в директорию, в которую ранее сохранили файл ping.pcap и с зажатым shift вызываем контекстное меню.



В появившемся окне терминала, выполняем серию команд для установки scapy:

```
sudo apt-get update
sudo apt-get install python-scapy
```

Клонируем репозиторий Net2axis:

```
git clone https://github.com/lucasbrasilino/net2axis
```

Конвертируем ping.pcap в ping.dat:

```
mv ping.pcap ./net2axis/tool
cd ./net2axis/tool
python net2axis.py -w 64 -i 100 -d 100 ping.pcap
```

Рассмотрим параметры скрипта:

`-w` ширина axi-stream;

`-i` задержка в тактах с момента снятия сигнала сброса до передачи первого пакета;

`-d` задержка в тактах между последующими пакетами;

В директории /tool появился файл ping.dat, который можно использовать при симуляции. Обзорный тестбенч находится в директории sim, сам генератор пакетов расположен в директории hdl. После запуска симуляции я получил такую временную диаграмму:

Расширяем функционал

Пока писал этот текст возникла идея реализовать скрипт с обратным функционалом(axi-stream → *.pcap). Первым делом добавим в симуляцию блок, который будет сохранять каждый принятый пакет в виде новой hex-строки. У меня получился такой код:

```
integer fdesc, i;

    initial begin
        forever begin
            fdesc = $fopen("outp.txt","a");
            while(!(M_AXIS_TREADY && M_AXIS_TVALID && M_AXIS_TLAST)) begin
                @(posedge ACLK) begin
                    if (M_AXIS_TREADY && M_AXIS_TVALID) begin
                        for (i = 0; i < 8; i=i+1) begin
                            if (M_AXIS_TKEEP[i]) $fwrite(fdesc,"%h",M_AXIS_TDATA[i*8+:8]);
                        end
                        if (M_AXIS_TREADY && M_AXIS_TVALID && M_AXIS_TLAST)begin
                            $fwrite(fdesc,"\n");
                            $fclose(fdesc);
                        end
                    end
                end
            end
            wait(!(M_AXIS_TREADY && M_AXIS_TVALID && M_AXIS_TLAST));
        end
    end
```

Теперь дело за малым - реализовать python-скрипт, который будет конвертировать полученный текстовый файл в формат *.pcap:

```
1. from scapy.all import *
2. text_dump = open('outp.txt', 'r')
3. for i in text_dump:
4. hex_str = i.split('\r\n')[0]
5. current_packet = Ether(hex_bytes(hex_str))
6. wrpcap('outp.pcap', current_packet, append=True)
```

Сохраняем скрипт в файл axis2net.py, размещаем его в одной директории с файлом outp.txt и запускаем командой:

```
python axis2net.py
```

После запуска должен появиться файл outp.pcap, который отлично открывается при помощи wireshark. Я использовал всё ту же scapy, на удивление эта библиотека оказалась простой и функциональной. На хабре есть хорошая статья про данную библиотеку - рекомендую ознакомиться.

На этой ноте завершаю свой обзор, благодарю за внимание!

P.S. Хочу обратиться к сообществу FPGA-Systems. На просторах сети много достойных проектов, о которых знают не все. Если вы нашли что-то полезное - напишите небольшой обзор, думаю многие будут благодарны.

Понравилась статья? Не забудьте поддержать автора. Задать вопрос автору можно **здесь**

Есть идеи для следующей статьи? Напишите нам fpga-systems@yandex.ru