RFLAN Documentation

Replicating RFLAN

The BytePipe x9002 Hardware Development Kits (HDKs) we received as a team came preloaded with an application called RFLAN. Developed by NextGenRF, RFLAN is a CLI tool that runs immediately after booting the BytePipe. This documentation follows the steps on how to build RFLAN as seen on NextGenRF's GitHub found here: https://github.com/NextGenRF-Design-Inc/bytepipe_sdk/blob/main/src/rflan/README.md

Here's the full description of RFLAN, as written on NextGenRF's GitHub:

> *The RFLAN application is an embedded C real-time application running on the Cortex R5 real-time processing unit (RPU). The programmable logic includes the ADRV9001 synchronous serial interface along with DMAs for streaming IQ data or modem data to and from shared memory. The RFLAN firmware configures and controls the ADRV9001. It also directs DMA data to and from shared memory to the command line interface (CLI) using a serial port or Ethernet. The following diagram shows the main components of the RFLAN including firmware running on the RPU and programmable logic. The RFLAN can operate as a standalone application or integrated with a Linux application running on the Cortex A53 using interprocessor communication (IPC).*

Building The Project

The first step in building RFLAN is to first build the source. Start by opening a terminal where you intend to build the project files. We'll start by cloning the BytePipe SDK repository and creating a workspace directory with the following commands:

```
git clone https://github.com/NextGenRF-Design-Inc/bytepipe_sdk.git
cd bytepipe_sdk/
mkdir workspace
cd workspace/
```

Next we'll be building the programmable logic. Before running the make command, we need to make sure we've sourced Vivado since that's the tool being used to build the HDL. To source Vivado, find the install location of Vivado and run the source command on the settings64.sh file. If you haven't downloaded the Xilinx software suite yet, see *Progress Documentation > Setting Up the Build Environment.* I installed my Xilinx tools within the tools/ directory in the root user directory, so my source command looks like this:

```
source /tools/Vivado/2021.1/settings64.sh
```

But the file location may be different depending on where your Vivado installation is located so pay attention to where Vitis, Vivado and Petalinux are being installed.

Once Vivado has been sourced, it's time to run the make command for building the HDL. This step will take a while, but depending on specifications of the system the build process can take more than an hour. Run the following make command to start the build:

```
make -f ../src/rflan/Makefile hdl
```

After the HDL build is over, it's time to build the RPU software for RFLAN. This build requires sourcing Vitis. Sourcing Vitis is similar to Vivado, as the only step is to run the source command with the path to settings64.sh in the Vitis install directory. On my system, the command would look like this:

```
source /tools/Vitis/2021.1/settings64.sh
```

Like with sourcing Vivado, the path to Vitis may be different depending on where Vitis is installed. The Makefile for the RPU software will generate the hardware platform, board support package, and RFLAN application. The project can even be opened using Vitis after the make command has completed and the project file to open RFLAN in Vitis will be found in workspace/rflan. Start the build with the following command:

```
make -f ../src/rflan/Makefile sw
```

The next step is to build the ADRV9002 profile, which can be done with another make command like so:

```
make -f ../src/rflan/Makefile profile
```

Deploying the Application to the BytePipe

After completing the necessary builds in the previous step, the project can now be packaged and deployed to the BytePipe. A Makefile can be run to generate the necessary contents for running RFLAN to be copied to an SD card. Running the following command will generate these files:

```
make -f ../src/rflan/Makefile sd_card
```

With an SD card handy, connect the SD card to the development machine and copy the files located in the sd_card directory to the SD card. Now, with the SD card inserted in the BytePipe, RFLAN will now run on boot!