FPGA Project: Startup Sheet

Jade Melliti and Alexandre Lamy

École des Mines de Saint-Étienne

1 Prerequisites

- Have a Python environment on your computer
- Have downloaded Vivado (version 2022.2)
- Have downloaded HTerm
- Have basic knowledge of Python and SystemVerilog
- Have the FPGA board (PYNQ-Z2)

2 FPGA Board Installation

The board needs to be powered/connected using a micro-USB port (highlighted in green in the image) to a computer. Additionally, the UART module needs to be connected to the FPGA board (via the PMODs) and to the computer (via another micro-USB cable).
Regarding the UART connection, it must be plugged into the "PMOD" section located on the right side of the PYNQ-Z2 board.

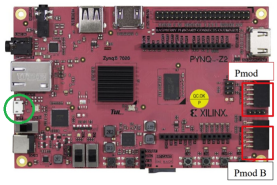

Fig. 1. PYNQ-Z2

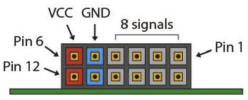

Fig. 2. Pin UART-PMODA

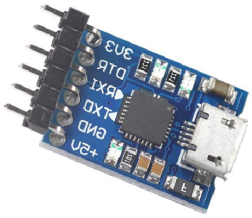

Fig. 3. UART Module

On this port, there are two possible levels for the connection. As we can see in our constraint file:

```
set_property -dict {PACKAGE_PIN Y18 IOSTANDARD LVCMOS33} [get_ports RTS_o]
```

The user manual allows us to say that Y18 corresponds to pin 1. Therefore, the UART must be connected to the upper part of the PMOD (see previous photo). The +5V of the UART module must be connected to port 6 (VCC), the GND to port 5 (GND), and so on.

3 Hardware Design on FPGA

3.1 Opening the Project

Open the Vivado software, open a project by clicking on "open project" in the "Quick Start" section. Then, select the .xpr file located in the folder corresponding to the project to be studied.

3.2 Code to Board Conversion

Once the project is open and the board is correctly connected to the computer, we need to convert our code, which is in SystemVerilog, into a binary file that contains all the necessary information to configure the FPGA. To do this, in the left menu of the Vivado window, in the flow navigator, find the "Program and debug" section. First, click on "generate bitstream". Once the file is generated, download it to the board with this binary file. This allows the FPGA to execute the designed hardware. To do this, simply right-click on open target and choose "autoconnect". Then, it is finally possible to program the board by clicking on "program device".

3.3 Serial Communication with HTerm

To verify the proper functioning of our program and therefore our FPGA, it is important to test our commands on a terminal (like HTerm) for serial communications.

Regarding our program, it is important to enter the different letters in hexadecimal. Several settings need to be adjusted on HTerm, such as connecting to the correct COM port, checking the baud value, choosing "LF" in the newline menu to have the response of each command on a new line. It is especially important not to forget to set the sent data type to "HEX". The following image shows the responses we obtained by sending the key (0x4B or 0x6B), the nonce (0x4E or 0x6E), the associated data (0x41 or 0x61) with padding (add 80 00), the wave (0x57 or 0x77) with padding (add 80 00 00), the go which triggers the ascon (0x47 or 0x67), the T (0x54 or 0x74) to get the tag, and the C (0x43 and 0x63) to get the cipher.

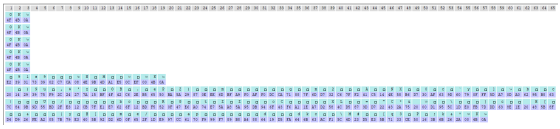

Fig. 4. HTerm Result

4 Interface and Acquisition

4.1 Serial Communication with Python

Another simpler way to establish serial communication with the board is to use a Python code. In our case, the file that allows this connection is the Term$_A$ $sconfile. Just enter the COM port on line 22 of the Python code and run the script. You will then need to enter al$

```
19  class LoadCheckParams:
20      @staticmethod
21      def port(port):
22          return port or "COM20"
23
24      @staticmethod
25      def baud_rate(baud_rate):
26          return baud_rate or 115200
27
28      @staticmethod
29      def timeout(timeout):
30          return timeout or 1
```

Fig. 5. COM Selection

```
Entrez le nombre D'ECG voulu : 4
Entrez le Nom du fichier Excel (Sans .csv) : waveform_example_ecg
Entrez la valeur de la Clé : 8A55114D1CB6A9A2BE263D4D7AECAAFF
Entrez la valeur du Nonce : 4ED0EC0B98C529B7C8CDDF37BCD0284A
Entrez la valeur de la Donnée Associé (Sans Padding) : 4120746F2042
```

Fig. 6. Python Terminal

The program will launch and display the filtered ECG curves with a blue cross representing the peak of ventricular contraction, which allows determining the beats per minute. The heart rate value will be the title of the curve in Python.

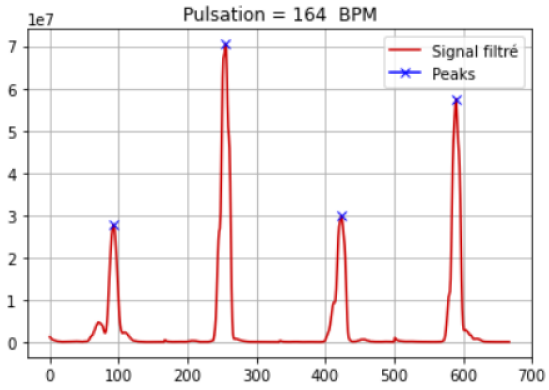

Fig. 7. Python Figure