FPGA-based electrocardiography (ECG) signal analysis system

Giancarlo Cincotti, Sagar Malhotra, Sara Quercetti
Laboratory of Advanced Electronics
Final Project

March 26, 2018

Abstract

The aim of our project is to develop an electrocardiogram using an FPGA. We acquire the body potential through two electrodes and make the signal readable by the ADC converter by an analog circuit. The FPGA filters the signal, detects the peaks and shows the heart rate on the display.

1 Signal acquisition and analog circuit

The signal is acquired using ECG electrodes on the wrists. Amuchina() gel or similar can be used to ensure contact and conductivity between the skin and the electrodes. The electrodes are connected to the circuit by cables shielded to ground, that reduce common mode noise. The analog circuit is shown in Figure 1.

The first part of the circuit is an instrumentation amplifier, composed of three different op-amps: two OP77 are used to buffer each input and one 741 is used as differential amplifier. R_G between the two inverting inputs of the buffers has the role of increasing the differential-mode gain of the buffer pair while leaving the common-mode gain equal to 1: this increases the common-mode rejection ratio (CMRR) of the circuit. Furthermore it allows the gain of the circuit to be changed by changing the value of a single resistor: for this reason we used a potentiometer for R_G, to be able to choose the best value for it during the signal acquisition and to fit it to the quality of the signal. The resistors used are: $R_1 = 10\,k\Omega$, $R_2 = 1\,k\Omega$,$R_3 = 100\,k\Omega$, R_G is a trimmer up to $5\,k\Omega$. The best result was obtained setting $R_G = 1.32\,k\Omega$. So the total gain of the whole instrumentational amplifier is

$$G = \left(1 + \frac{2R_1}{R_G}\right)\frac{R_3}{R_2} \simeq 1600$$

After the instrumentational amplifier there is an analog high pass filter, that was essential to cut off a very instable DC component in the signal. We used the components $C_1 = 10\,\mu F$

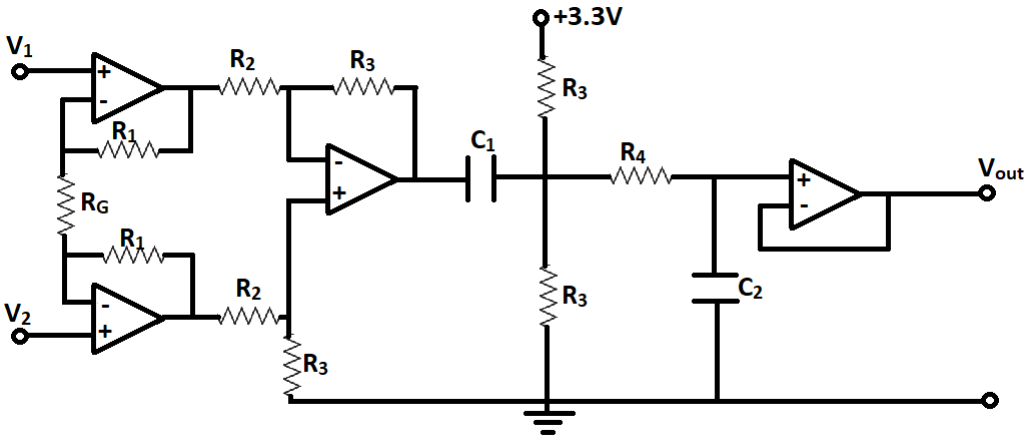

Figure 1: Scheme of the analog circuit

and $R_3 = 100\,k\Omega$ giving a cutoff frequency of $f_{ahp} = \frac{1}{2\pi R_3 C_1} = 0.16\,Hz$. On the output of the high pass filter we added a DC voltage of $1.65\,V$ through the $3.3\,V$ power supply of the FPGA and another R_3 resistor to divide the voltage. In this way the signal is centered in the ADC voltage range needed by the FPGA.
Then we added also an analog low pass filter to cut off the $50\,Hz$ noise that was still too high with respect to our signal. $R_4 = 50\,k\Omega$ and $C_2 = 100\,nF$ have been used to obtain a cutoff frequency of $f_{alp} = \frac{1}{2\pi R_4 C_2} = 31.8\,Hz$. At the end we added a buffer made with an OP77 op-amp to avoid the impedance matching with the FPGA ADC, because the FPGA has very low input impedance. All the op-amps in the analog circuit are alimented by two $9\,V$ batteries, to avoid using a power supply in the same circuit of the human body and adding further $50\,Hz$ noise in the signal.

2 Digital signal processing

The signal given by the analog conditioning circuit is acquired and processed using FPGA Spartan 3AN board by Xilinx. The written program uses FPGA ADC, DAC and LCD to, respectively, acquire the signal, output the conditioned signal and show the heart rate in beats/minute.
The main structure of the program is shown in the block diagram of Figure 2. In the scheme the gray arrows represent binary arrays, while black thin lines are referred to 1 bit. All the modules of the program are synchronized with the $50\,MHz$ quartz clock.

2.1 Description of blocks

- **ADC Acquisition**: the ADC board acquires 2 channels 14 bits data at the sampling frequency of $50\,kHz$. Channel A acquires the output signal of the analog circuit, while

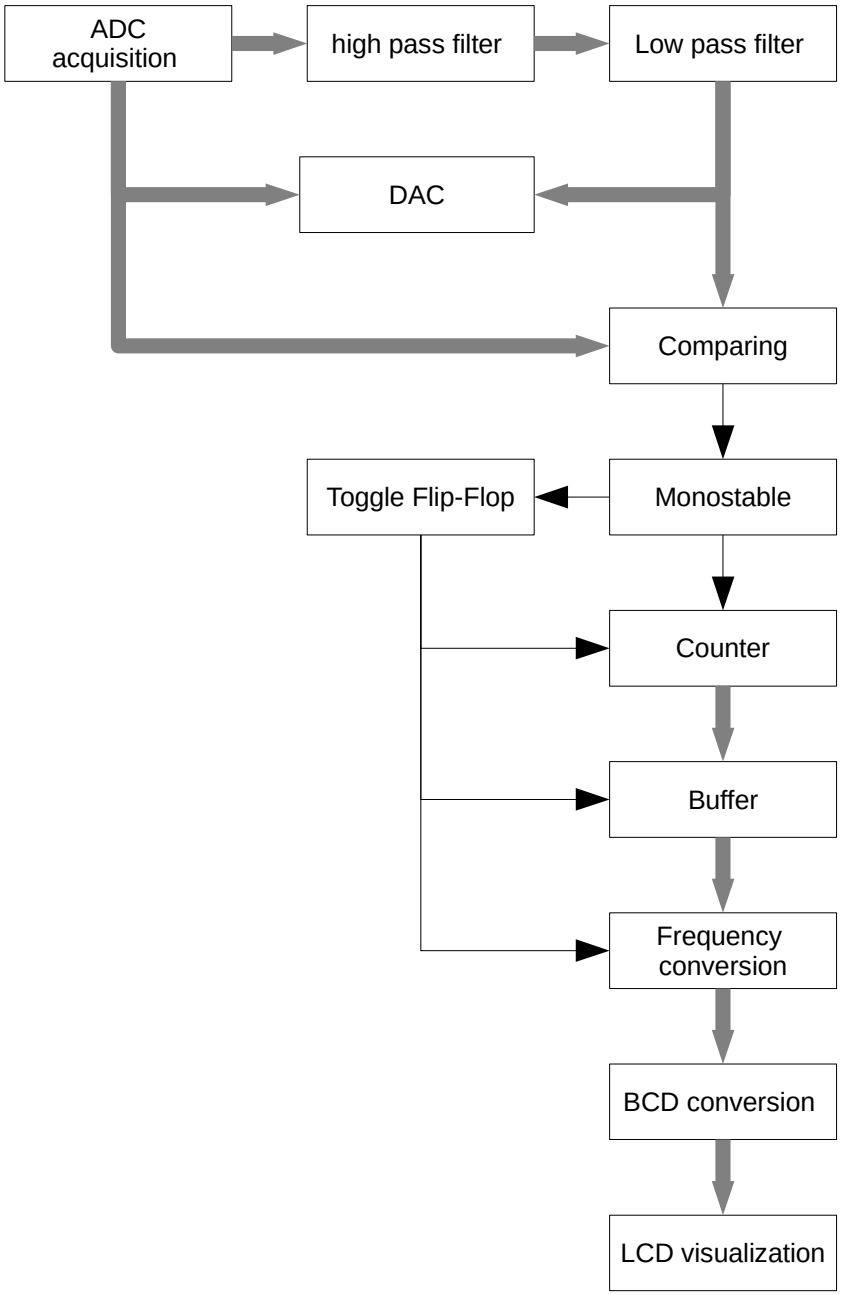

Figure 2: Block diagram of the Verilog program.

on channel B we put a DC signal from the power supply that we need as threshold to detect peaks in the signal. The digital signals are placed in specific registers.

- **High pass filter**: it cuts off the residual DC components. The cutoff frequency set is $f_{dhp} \simeq 3.9\,Hz$. The implementation of the filter is described in section 2.2.

- **Low pass filter**: it filters the residual high-frequency components over the cutoff frequency of $f_{dlp} \simeq 31.1\,Hz$ including the $50\,Hz$ oscillation still present in the signal after the analog filtering. The result is smoothing the signal.

- **DAC**: the DAC outputs in analog the filtered heart signal and the threshold, to make them displayable on the oscilloscope.

- **Comparing**: this block takes the filtered heart signal and the threshold value to compare them, giving a 1 bit signal when the heart signal is over threshold.

- **Monostable**: the rising heart beat sets the Monostable to 1. The monostable is then set high for a time period $\tau = 100\ ms$. This time has been chosen because it is in any case less than time between one peak and another (max HR 600 beats/minute) but it is sufficient to avoid secondary peak detection.

- **Toggle flip-flop**: the monostable signal is then fed into the toggle flip-flop, which changes state at each positive edge of the monostable, that is at any peak detection. The behavior of monostable and toggle flip-flop with the signal is schematized in Figure 3.

- **Counter**: this block is done with two 8 bits counters in series. It counts the time period for which the toggle flip-flop is set 1. This is indeed nothing but the time period of the heart beat. The counting frequency is $1\,kHz$, so the output is in milliseconds. It has been chosen in order to set the error on the HR detection, in the worst case assumed as 40 BPM, lower than 1 BPM, so it is possible to show the correct value on the display. The counter resets its value to 0 at the negative edge of the monostable when the toggle's value is 0.

- **Buffer**: it takes the data from the counter and it locks it until the clock changes. The buffer is fed by the negation of the toggle flip flop signal, so its output is just the time period between two consecutive peaks.

- **Frequency conversion**: the associated module converts the time period given by the buffer to a binary number corresponding to the hart rate in beats/minute. The block must do a division between the value 60'000 (1 minute in milliseconds) and the period. We calculate the number of beat periods that exist in 1 minute by consecutively subtracting the beat period from 60'000 till the remainder is less than the given beat period and counting how many times the subtraction is done. The number of possible consecutive subtractions is our HR value in BPM.

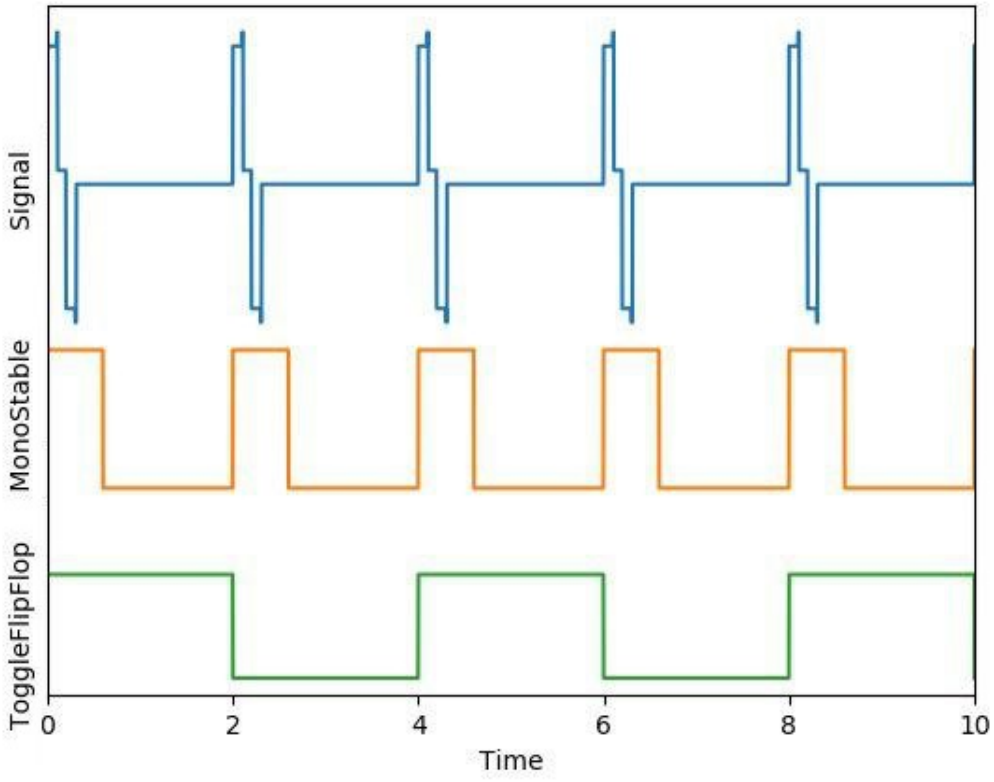

Figure 3: Signal conversion in FPGA

- **BCD conversion**: it takes the number in binary and converts in BCD code.
- **LCD visualization**: this module takes the HR value in BCD and shows it in the LCD display in decimal units.

2.2 High Pass Filter

We developed the high pass filter starting from the transfer function $H(s) = \frac{s\tau}{1+s\tau}$ and the bilinear transform $s \to \frac{2}{T}\frac{z-1}{z+1}$, where T is the sampling time. We obtain the simulator

$$V(z) = \frac{2\tau}{T+2\tau}\frac{z-1}{z-c}$$

with a zero in $z = 1$, corresponding to 0 frequency, and a pole in $c = \frac{\frac{2\tau}{T}-1}{\frac{2\tau}{T}+1}$. Let's call $\alpha = \frac{2\tau}{T+2\tau}$. We obtain the equation

$$Y(z)\left(1-\frac{c}{z}\right) = \alpha\left(1-\frac{1}{z}\right)X(z)$$

corresponding to the difference equation

$$y[n] = \alpha(x[n] - x[n-1]) + cy[n-1]$$

Our simulator corresponds, in Laplace domain, to

$$V(s) = V(z = e^{sT}) = \alpha \frac{e^{sT} - 1}{e^{sT} - c} \simeq \alpha \frac{sT}{1 - c + sT} = \frac{\alpha}{1 - c} \frac{sT}{1 + \frac{sT}{1-c}}$$

so, in analogy with the high pass transfer function, we can write our $\tau = \frac{T}{1-c}$.
Now let's define $\epsilon = \frac{T}{2\tau}$: we can write $\alpha = \frac{1}{1+\epsilon}$ and $c = \frac{1-\epsilon}{1+\epsilon}$. We see that $1 - c = \frac{2\epsilon}{1+\epsilon} << 1$, so we can take $c = 1 - 2^{-k}$. It follows that $\alpha = \frac{1}{1+\epsilon} = \frac{c+1}{2} = 1 - 2^{-(k+1)}$.
So our difference equation can be implemented as

$$y[n] = (1 - 2^{-(k+1)})(x[n] - x[n-1]) + (1 - 2^{-k})y[n-1]$$

The cutoff frequency is $f_{3dB} = \frac{1}{2\pi\tau} = \frac{1-c}{2\pi T} = \frac{2^{-(k+1)}}{\pi} f_s$, where $f_s = 1/T$ is the sampling frequency, and can be chosen changing the parameter k.

3 Results

At the end of the digital filtering processes, the DAC output of the FPGA shows very good and precise ECG signals. One example is shown in Figure 4. The HR displayed in the LCD are in perfect agreement with the period measurable by the oscilloscope, proving the correct work of the frequency calculation. The program could be improved by establishing a definitive value for the threshold with which comparing the signal, instead of adjusting it each time with the power supply, but the low number of tests that we could effectuate didn't gave us the possibility to determine an universally acceptable value.

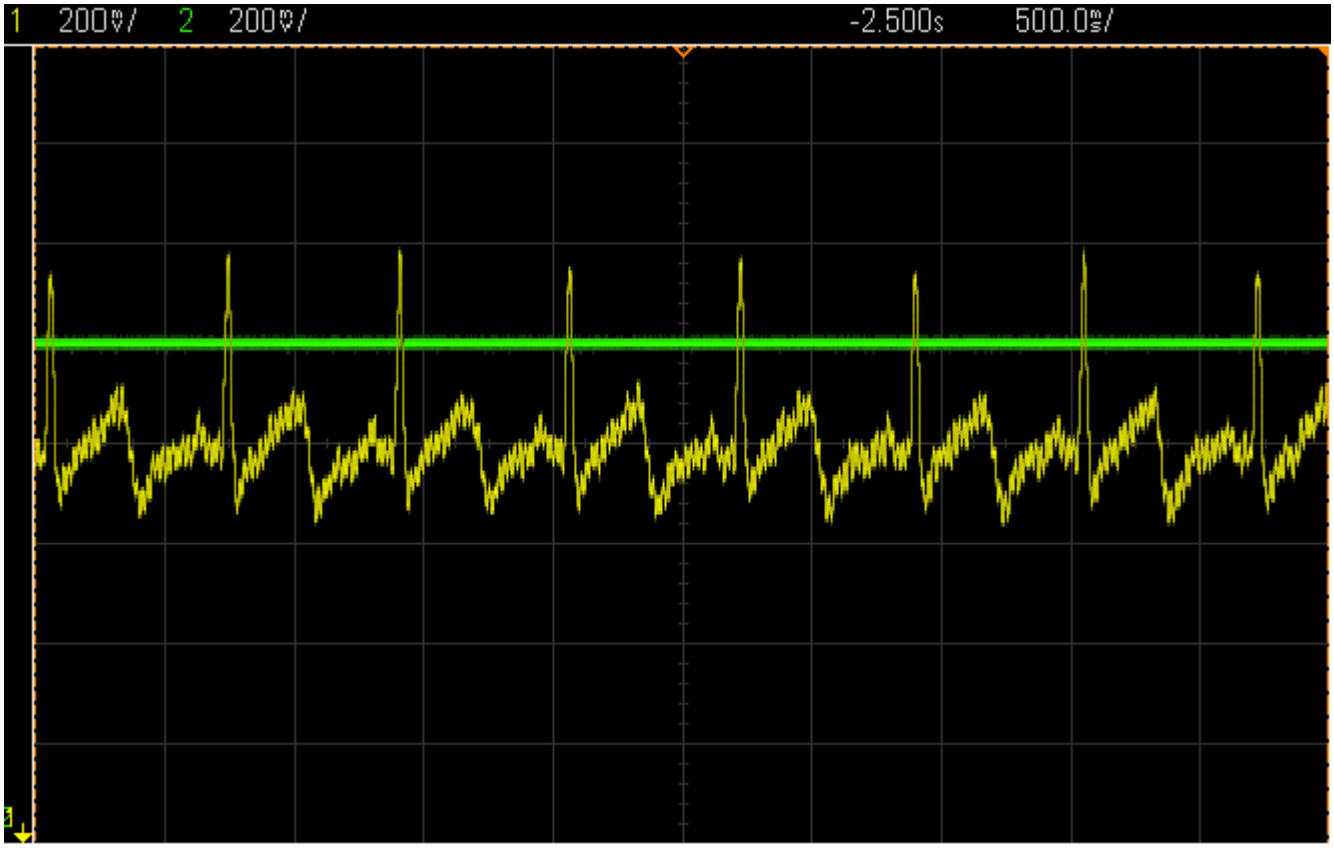

Figure 4: ECG signal obtained. The green line represents the adjustable threshold used to detect the peaks.