Final Project Report:

Wireless Musical Electrocardiogram

Amy Tang and Sinit Vitavasiri

Group 12

Abstract

This project aims to implement a wireless electrocardiogram system with music applications. Three-lead sensors take an electrocardiogram reading from the heart. The data from the sensors are wirelessly transmitted to a digital system that displays a filtered, electrocardiogram signal from the patient onto an oscilloscope. The system detects the heartbeat of the patient from the data and produces a rhythmic indicator with numerical display, as well as an abnormality alarm. The heartbeat signal also drives the tempo of a pre-stored music piece.

TA: Hyunjoo Jenny Lee
Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.111 – Introductory Digital Systems Laboratory

WIRELESS MUSICAL ELECTROCARDIOGRAM

Overview

The integration of wireless communication into medical applications is continuing to revolutionize hospital care. With wireless sensors, patients are no longer bound to a bed where the medical instruments monitor their vital signs. Wireless communication for medical applications is able to solve clinical needs and risks, while providing the patient with the freedom of movement.

The concept of this project has vast potential for further extensions and advancements. Real-time automated medical monitoring using small wireless sensor nodes will redefine not only hospital care, but also work, home and recreational activities.

The objectives of this project are: 1) to obtain electrocardiogram measurement from patient, 2) to intelligently analyze, store, and transfer data to end users, and 3) to be able to detect and report varying conditions of patient.

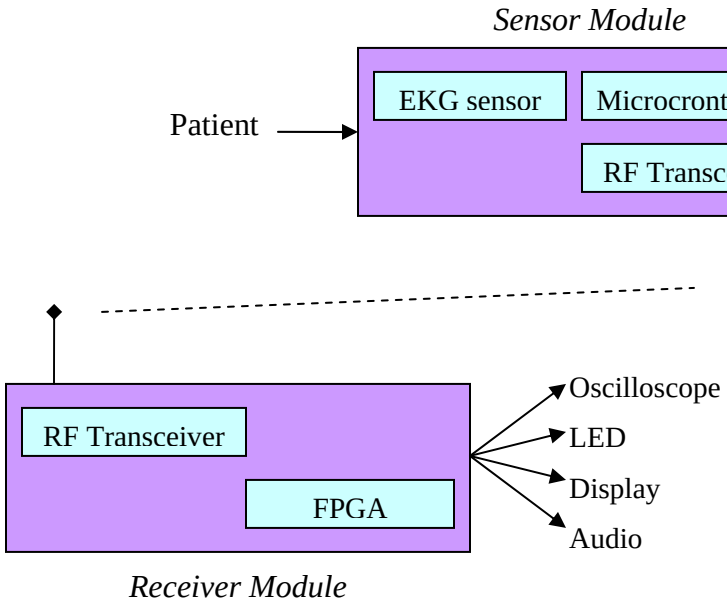

Figure 1: System block diagram

Project Description

The Wireless Musical Electrocardiogram system consists of the sensor and the receiver modules. The electrocardiogram (EKG) sensor takes an electric signal from the heart. The data obtained from this sensor is then digitized and intelligently transferred over the wireless network to the receiver, as shown in Figure 2. The wireless RF transceivers used in this project are ChipCon 1010 radios. The signal is stored and analyzed in the FPGA.

Because the EKG signal measured from patients is often noisy, the wirelessly received signal has to be filtered to remove baseline fluctuations as well as to attenuate

high-frequency noise. After noise reduction, the signal is converted from digital to analog format. The end user (e.g. nurse or doctor) is able to monitor the medical condition of the patient on the oscilloscope.

The filtered, digitized EKG signal is also input to the heart beat detector module. This module produces pulses when it detects a sharp peak of the EKG signal. The heartbeat pulses go to the external LED, which blinks according to the patient's heart rate, and to a module that determines the number of heart beats in one second. This three-digit number display gets updated every ten seconds. Moreover, in the situation where a patient's medical condition is dangerous, such as the heart rate is abnormally low or high, the abnormality-detector LEDs illuminate.

The Tempo Controller module changes speed of the music according to the patient's heart rate, i.e. the pre-stored music is sampled at a faster rate when the heart rate dramatically increases.

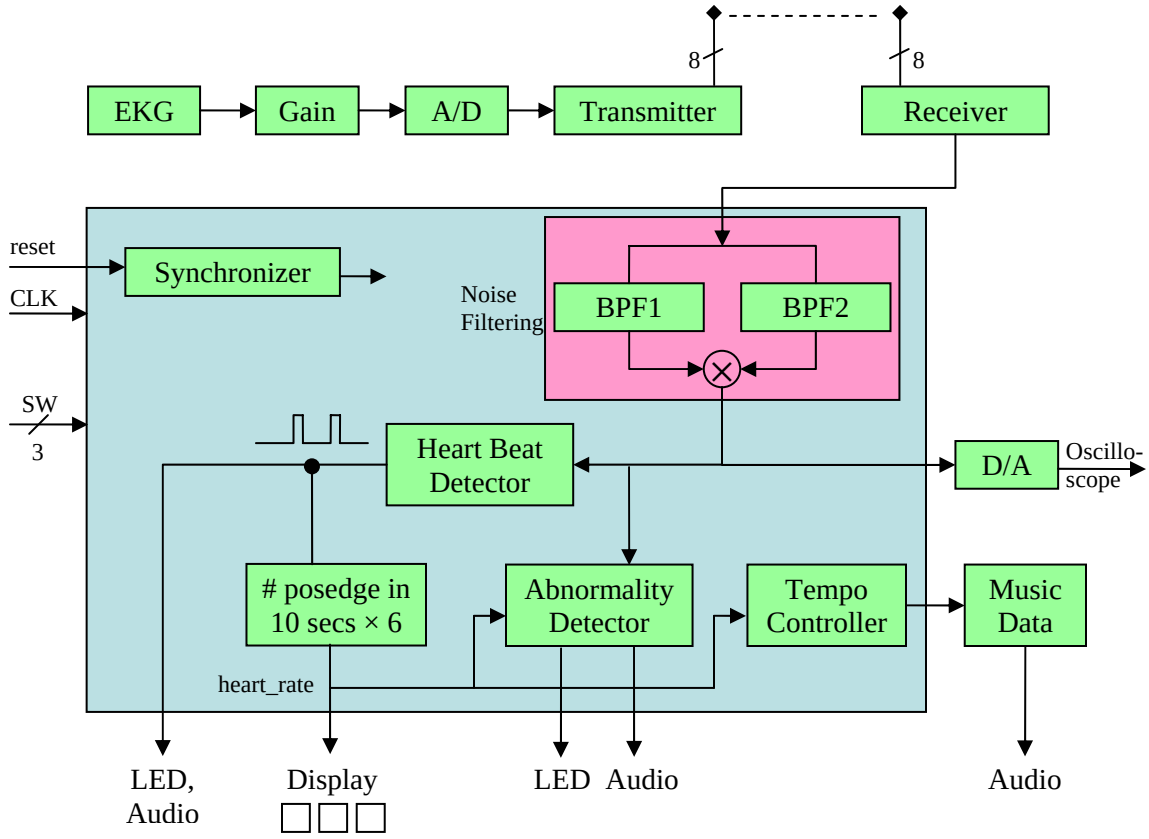

Figure 2: Detailed block diagram of the overall system

The reset signal stops the operation of the system and initializes all internal parameters. There are also three switches which select modes the user wants to operate. The three modes of operation for this system are:

Mode 1

- EKG Wireless Monitor
- Bright LED "Beat" Indicator
- Digital Heart Rate Display

Mode 2
- Abnormality Alarm
- Audio Heart-Beat Indicator

Mode 3
- Music of the Heart (Heart-Rate-Controlled Music)

Module Description/Implementation

Wireless Transceivers

The CC1010 Wireless Kit consists of an 8051 Compatible Microprocessor, 32 KB Flash Memory, memory, 10-bit ADC chip, and 8-bit, 300-1000 MHz radiofrequency transmission. The microprocessor is located on the CC1010EB Evaluation Module (fig.3). Computer language C is used to program the microprocessor, Hyper Terminal ("printf" function), RSS-232 UART port ("putchar" function), and LEDs. The EPROM Programming Compiler, Keil µVision2, is used to build the C hex files onto the CC1010EB through the CC1010 Evaluation Board (fig.3). Calibration of room temperature and measuring conditions can greatly affect the accuracy of wireless transmission.

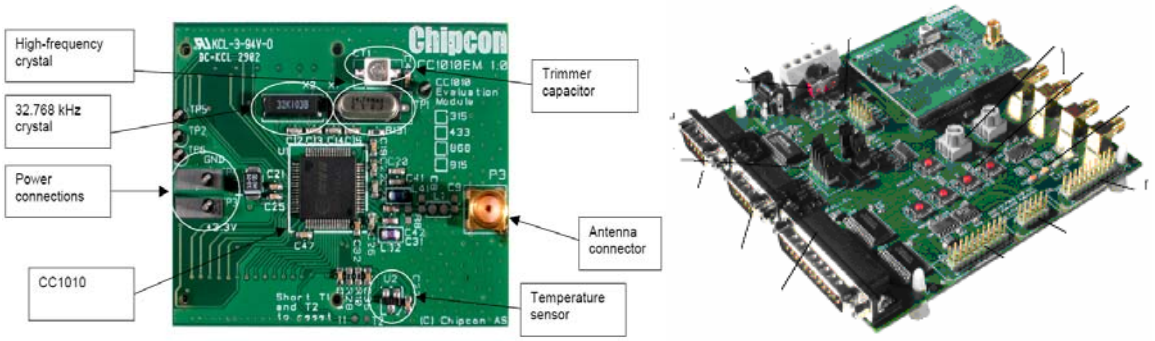

Figure 3: CC1010EM Evaluation Module (left) and Evaluation Board (right) (Courtesy of 2000-2005 Chipcon, All rights preserved by SmartRF, www.chipcon.com)

The analog EKG wave (700 mV VPP) is converted by the A/D converter into 10 bits of digital information. The two least significant bits are removed and bundled together in a package (up to a maximum of ten 2-bytes in a package), which includes a start and stop bit (a parity bit was left unused). This package is transmitted serially from an antenna and received serially on the receiver antenna. The bits are sent into the RSS-232 UART (Universal Asynchronous Receiver-Transmitter), which is standard for serial binary data interconnection between a port terminal and a data-communicating equipment.

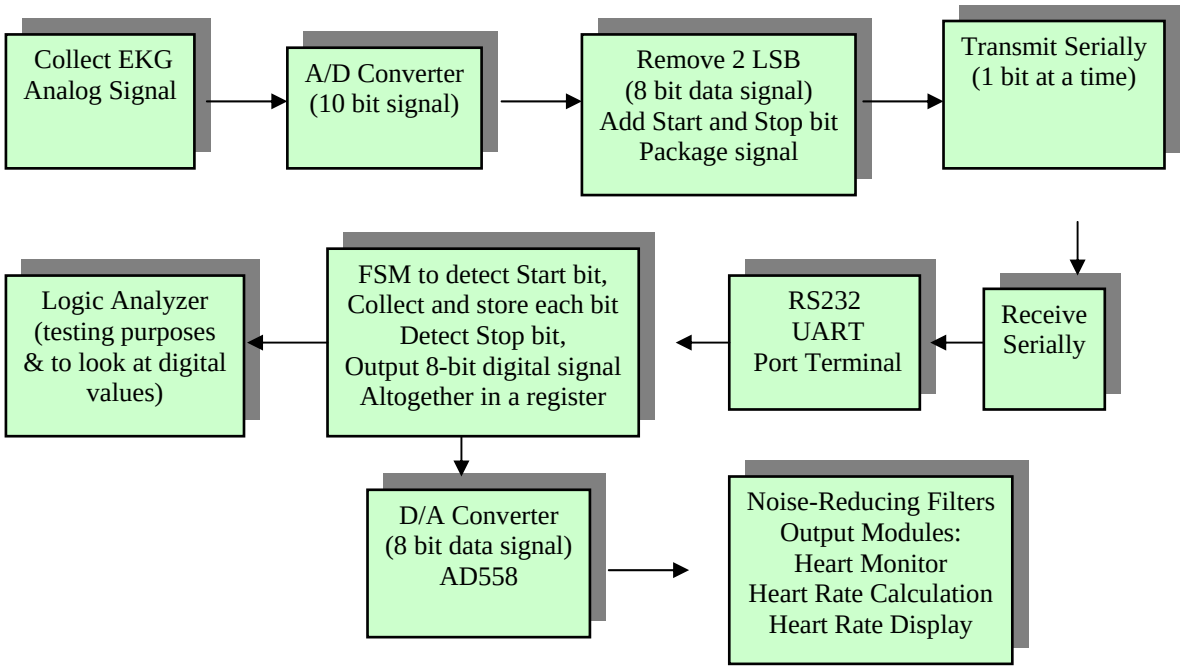

Figure 4: Control flow for the Wireless mode

A RSS-232 FSM is programmed into the lab kit to detect the start bit, extract the 8 bit signal, detect the stop bit, and send the EKG digital signal to the rest of the Wireless Musical Electrocardiogram system (fig.4). The RF sampling frequency is set to 915MHz, baud rate (bits/sec) at the terminal is set to 57k, and the frequency across the air is measured to be 12.5Hz. A block diagram of the wireless system can be found in Figure 5.

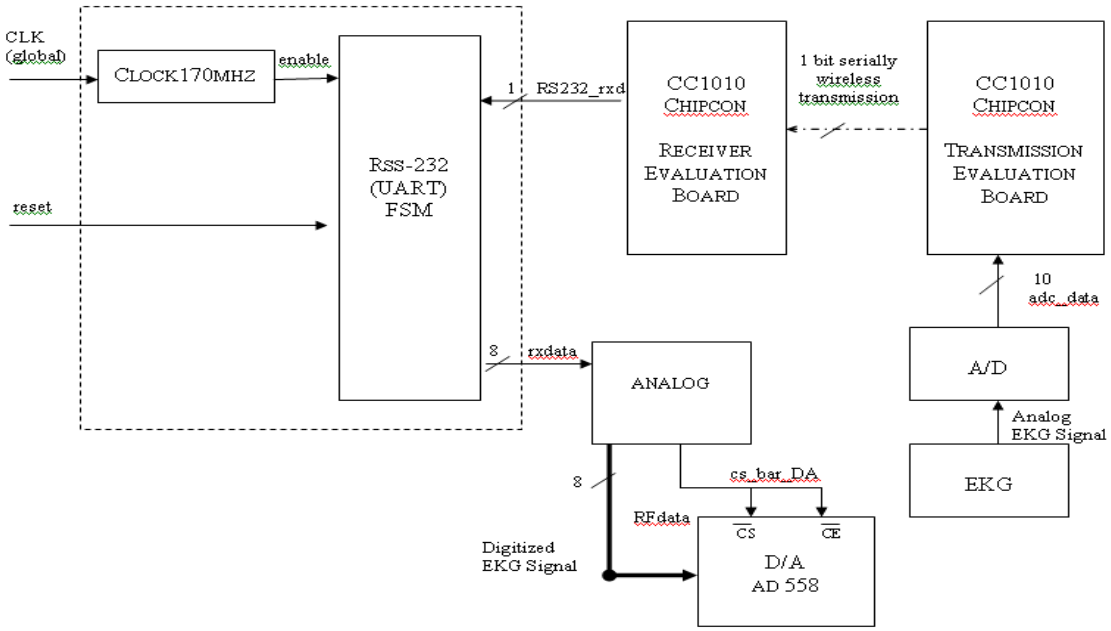

Figure 5: Block Diagram of the Transceiver (Transmittance and Receiving) System

Noise Filtering and EKG Wireless Monitor

Figure 6 shows the control flow of the system. First, the module is initialized when the *reset* signal goes low. Then it waits until the next sample (sample is a one cycle pulse every sampling period). The module receives the EKG signal and samples the output to the digital-to-analog converter at an equal rate of 250 Hz, i.e. the *sample* signal goes high 250 times per second. Note that the wireless receiver outputs the signal in digital format. The previously computed output signal sample is then output to the D/A converter (AD558). After the wirelessly received signal is stored in the SRAM, the system starts convolving and multiplying signals in order to reduce noise. Finally, it loops back and waits until it is time for the next sample.

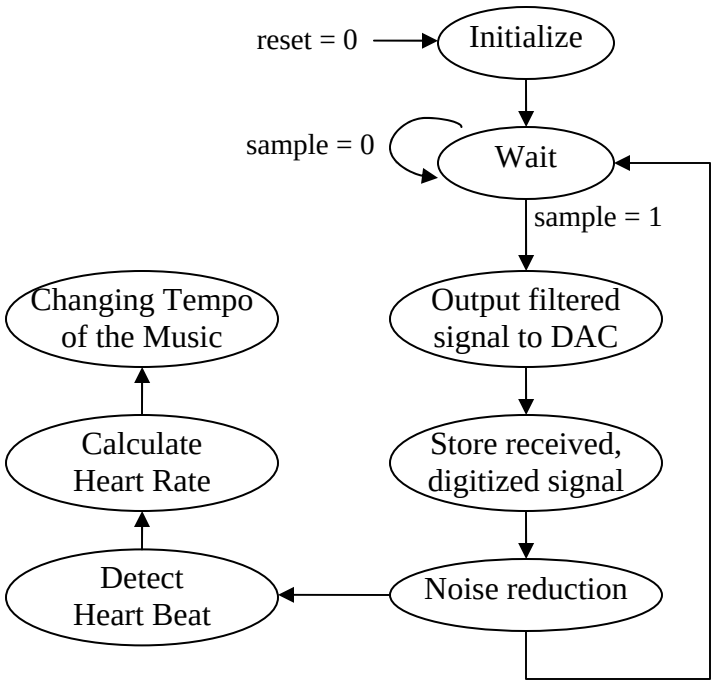

Figure 6: Control flow for the system

This subsystem for noise filtering consists of a major and two minor finite state machines. The major FSM controls the operation of the minor FSMs by sending start signals to each module in an appropriate time. The *analog* module reads the input and saves it to the SRAM at a rate of 250 Hz. This module also sending filtered signals (*int_data*) to the digital-to-analog converter at the same rate.

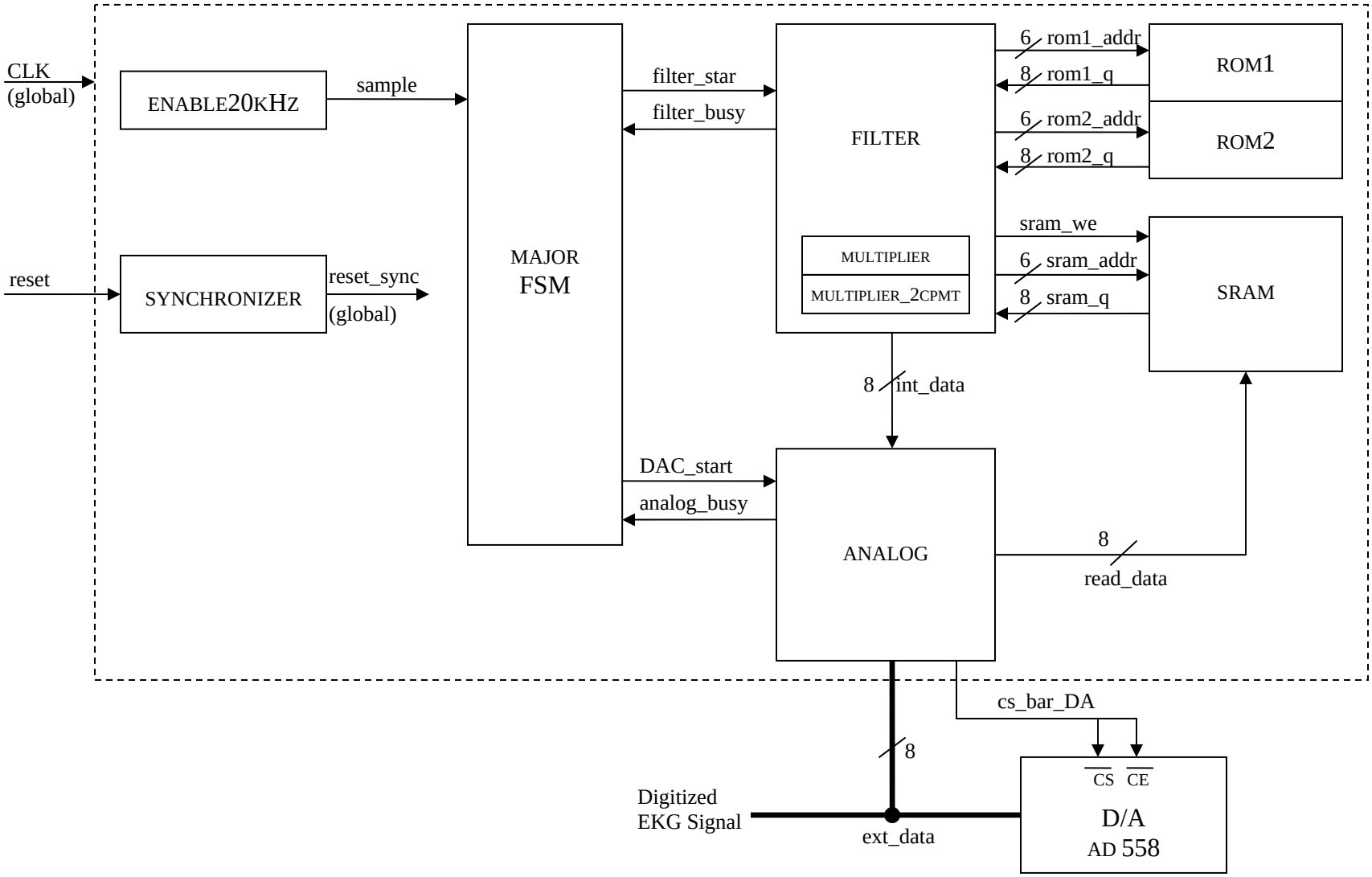

Figure 7: Block diagram of the Noise Filtering and EKG Wireless Monitor module

The *filter* module takes in the stored data in the SRAM and convolves it with the data from the two ROMs. The first ROM contains the impulse response of the first 64-tap bandpass filter whose band is from 3 Hz to 22.5 Hz. The second ROM contains the impulse response of the second bandpass filter whose band is from 3 Hz to 10 Hz. The module multiplies the two filtered signals and sends the result to the *analog* module to output the signal.

Digital Signal Processing: Noise Reduction

In order to reduce high-frequency noise, we need to select appropriate cutoff frequencies such that most of the signal energy is preserved. The clinically relevant energy in an EKG signal falls between 0.05 and 50 Hz. Figure 8 shows the block diagram of the noise filtering module. Note that the sampling rate is 250 Hz.

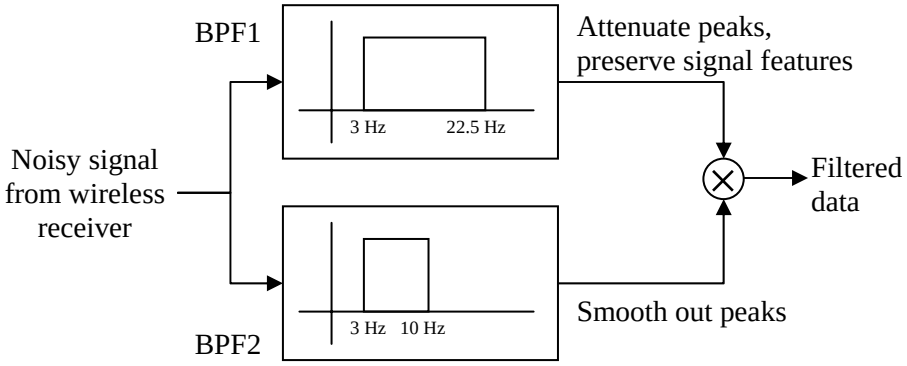

Figure 8: Noise filtering

The data is filtered with two different bandpass filters in parallel and the outputs are multiplied at the end. Multiplying the filtered signals in time domain allows the peaks of the EKG signals to amplify, while attenuating low amplitude noise.

The two 64-tap FIR filters are implemented in Matlab. Each filter has an order of 63 and is implemented by using Hamming window. The impulse responses of each filter are quantized to 8-bit values. The data for the FIR filters, saved in the .coe files, is then used to initialize the ROMs.

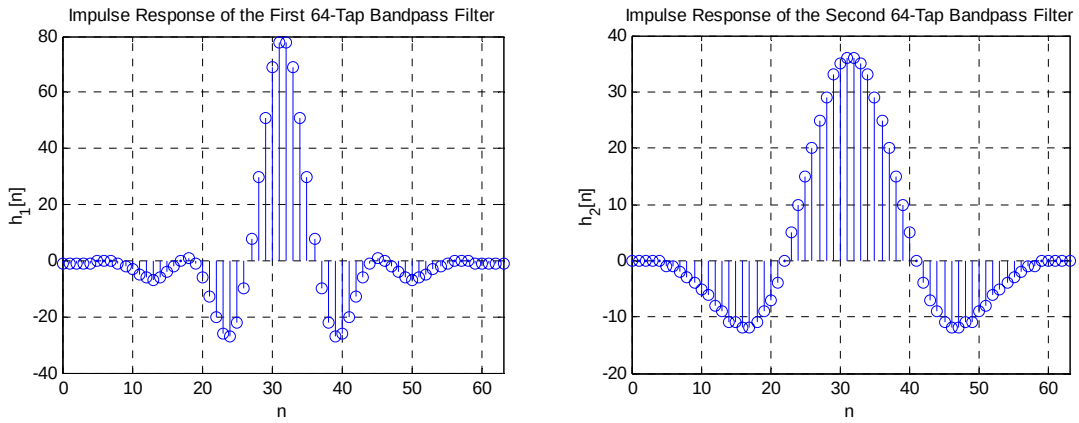

Figure 9: Impulses responses of the two bandpass filters

Bright LED "Beat" Display and Heart Rhythm Sound

The Heart Beat Detector module takes the filtered EKG signal as an input, and outputs the heart beat signal. At every positive edge of the clock signal, the module checks whether the input signal exceeds the pre-determined threshold voltage. If it does, the output goes high; otherwise, the output stays low. Consequently, the heart beat signal is a train of pulses. The heart beat signal is output to an external LED so that the LED illuminates according to the heart rhythm. The heart beat signal is also input to the *beep* module. If the user turns on the switch that enables the heart rhythm sound (see Appendix A for switch assignments), the system will beep whenever it detects a peak in the filtered EKG signal.

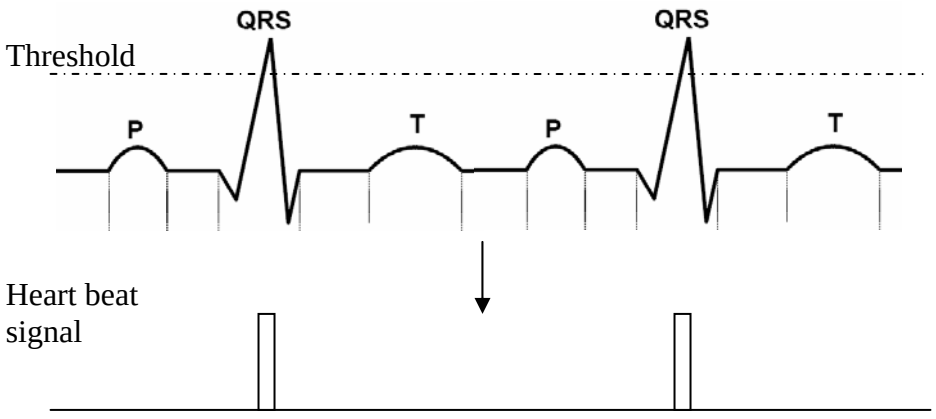

Figure 10: Determining the heart beat signal from the EKG

Digital Heart Rate Display and Detecting Abnormality

Heart rate is the number of heart beat per minute. To determine the heart rate, the module counts the number of positive edges of the heart beat signal in 10 seconds. The heart rate is obtained by multiplying the number of counts by 6. The heart-rate signal is input to the *bin2dec* module, which converts the heart rate into a three-digit decimal number. The three digits are sent to the alpha-numeric display.

If the heart rate falls below 50 or rises above 200 beats per minute, the external LEDs will blink and the alarm will go off (unless the user turns off the switch that enables this function), warning that the heart rate of the patient is abnormal.

Heart-Rate-Controlled Music

A classical music piece is sampled at the rate of 11.025 kHz. The samples are quantized to 8-bit digital data and then stored into the Flash ROM. A set of finite state machines controls the sampling play back of the music through the Audio Codec at the same sampling frequency. The LM4550 Audio Codec is an appropriate chip to use, due to it's DAC, ADC, headphone, and sampling rate conversion capabilities. Due to the sampling rate of the .wave file of the music (11.025 kHz), it is crucial that the sampling rate of the playback system matches this rate. Note that the DAC sampling rate for the audio system has to be specified by assigning the rate to register *2C* of the Codec.

An important concept to consider is increasing the tempo without changing the pitch. The changes of tempo for the music are set so that the tempo change will be distinguishable, but the change in pitch will be so miniscule that the average person will not be able to hear the difference. The increase in tempo corresponds to playing back less of the stored music digital data, while, the base tempo will be the playing back of every piece of stored music.

The digital system for changing tempo of a music piece at 11.025 kHz is implemented by playing a set of 1,764 consecutive stored data, skipping a certain number of samples, and then playing another set of 1,764 samples. The number of samples to be removed and the corresponding speed of the music are shown in the table below.

Heart Rate	Tempo of the Music (× tempo of the original piece)	Number of Samples to be removed after playing consecutive 1,764 samples
0-79	1	0
80-99	1.25	352
100-119	1.67	705
120-139	2	882
140+	2.5	1,058

Flash ROM

The music is extracted from a .wav file and converted into coefficients .coe files. The coefficients took the form of two bytes, and approximately 600,000 coefficients needed to be saved. It appeared most efficient to save these samples into the Flash ROM, which needed to written and programmed once. If the FPGA ROM is used, it would require constant reprogramming when any new module is added to the FPGA. To reprogram 600,000 coefficients into the FPGA ROM would require 3-4 hours every time. Programming the Flash ROM would require 5-6 hours once.

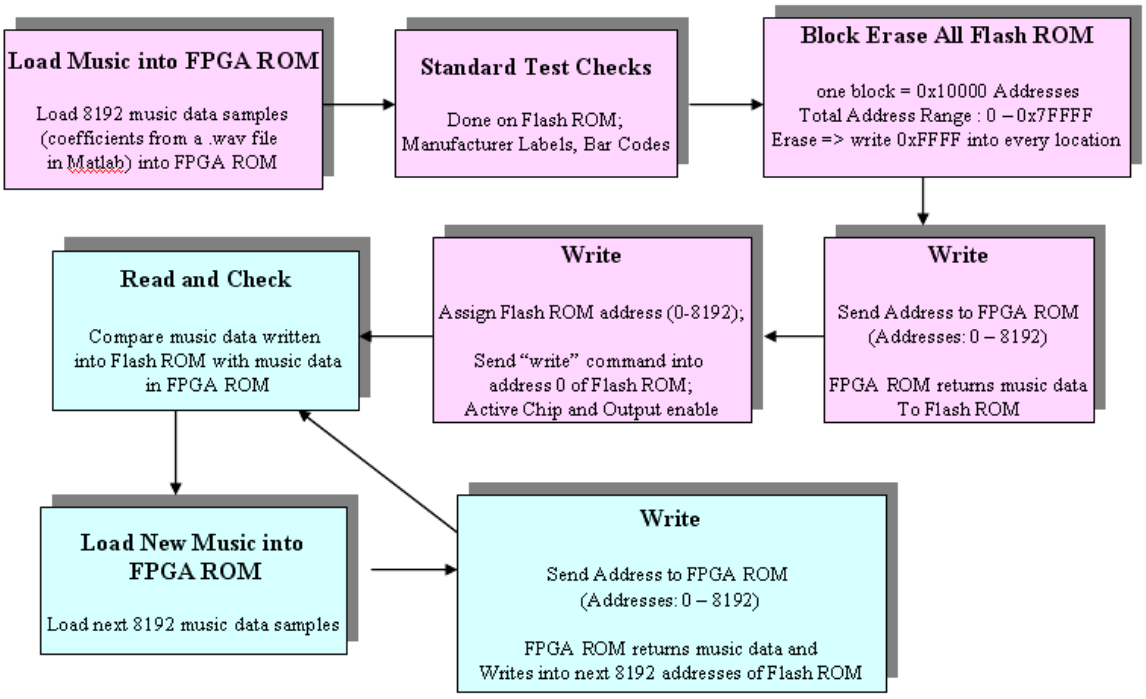

Figure 11: Flow Chart of the Flash ROM and FPGA ROM module

The 28F128-128Mbit Flash ROM is used in this system. This is an enormous amount of memory space; the 28F128 Flash ROM will be able to accommodate the 600,000 music samples. One problem is that the Flash ROM does not interface with .coe files. The .coe files interface only with the ROM in the FPGA. The solution is to temporarily save 8,192 coefficient samples into the FPGA ROM and then use a FSM to control the trafficking of reading the data from FPGA and writing them into the Flash ROM. New sets of 8,912 samples are saved into the FPGA ROM and then written into the correct addresses of the Flash ROM.

It is important to note that Flash ROM performs block erasing and not individual address erasing. The Flash ROM "erases" by writing 0xFFFF into the memory locations. If another data piece of data is written into a location, the Flash ROM will append the data together. If a location reads "0x0000", it actually can be re-written into.

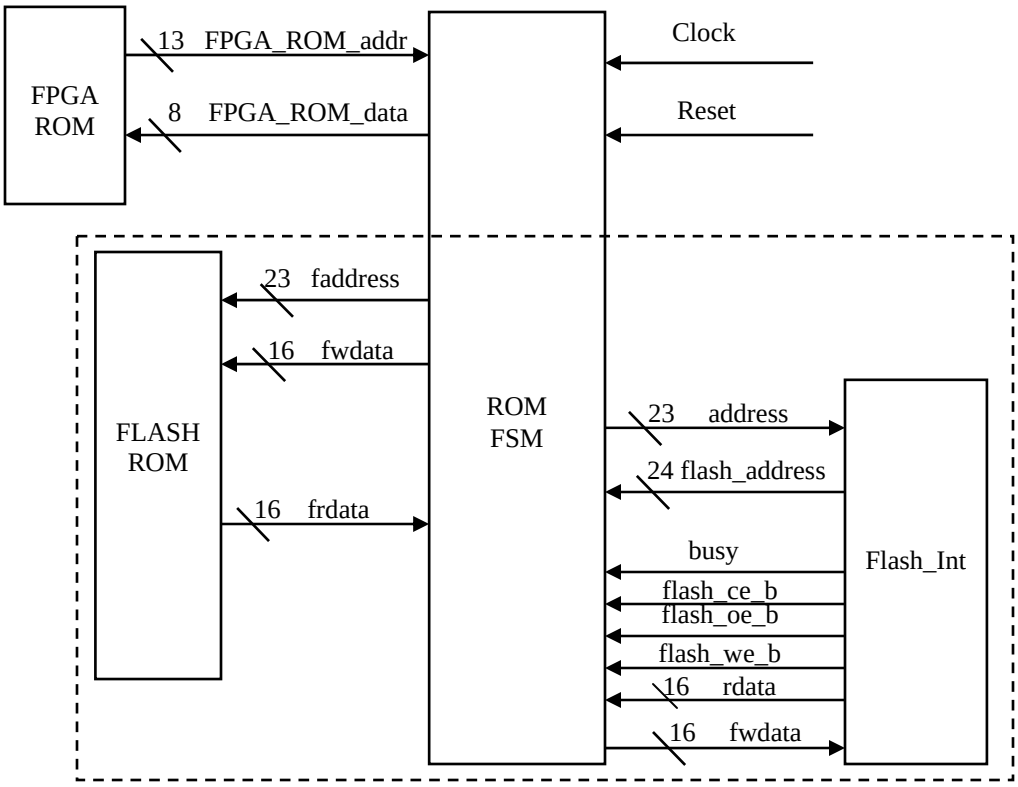

Figure 12: Block Diagram for programming the Flash ROM

Simulations

Below is an Altera Waveform, which exhibits the behavior of the RS232 FSM. The incoming serial bit information is rs232_rxd, which are being stored respectively into the b0, b1, …, b7. The 8-bit digitized EKG data output is `[7:0]rxdata`. The `[7:0]rxdata` register holds the current EKG value until the next EKG data signal is completely received.

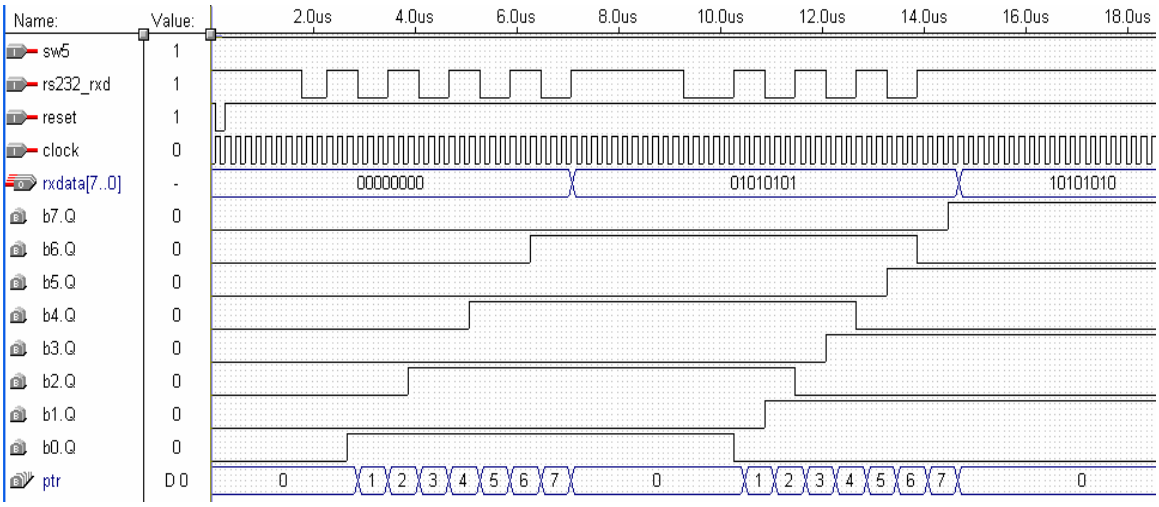

Figure 13: Wave form of the RS232 FSM

Testing/Debugging

Each subsystem/module needs to be tested extensively to verify its functionality. Various types of EKG sample signals, such as an abnormal, a noisy, or a heart-rate-varying EKG signals, are stored in the ROM. Those signals are used to test the system's functionality when the EKG signal deviates from the normal condition. Below are the lists of testing steps to be used for testing each subsystem.

EKG Front-End Sensor

- Produce EKG waves with correct frequency
- Produce EKG waves with the correct EKG shape in the 0-800 mV range

Wireless Module

- Correctly transmit and receive information between the transceivers
- Test the operation of the A/D converter on the CC1010
- Send a sine wave to the transmitter and observe the output signal of the receiver
- Send the EKG signal to the transmitter and receive

EKG Wireless Monitor

- Test the noise filtering process in Matlab by using sample EKG signals and verify that the noise is attenuated.
- Perform the noise filtering operation by using EKG signals sampled by the analog-to-digital converter (AD670).
- Perform noise filtering operation by using clean and noisy pre-stored EKG signals.
- Perform the noise filtering operation by using wirelessly transmitted EKG signals from the wireless module.

Heart Rate Output Module

- Verify that the heart beat signal is correct
- Display the heart rate on the three-digit digital display and verify that it gets updated every 10 seconds.
- Input normal and abnormal pre-stored EKG signals to the system and verify that the external LEDs illuminate if the heart rate per minute is not in the normal range.

Heart-Rate-Controlled Music

- Record/Save data into the Flash ROM and read back data
- Record/Save music file into the Flash ROM and read back data
- Play back stored music into speakers
- Change tempo of a song by inputting pre-stored heart-rate-varying EKG signals to the system.

In the testing process, due to limited resources, the actual Electrocardiogram (EKG) equipment produced very bad heart waves. Therefore, noisy EKG values from a health website are used to display the filtering function along with the other output modules. If an accurate EKG is connected to the system (and the signal is in the 700 mV range), it is believed that the Wireless Musical EKG would work correctly.

For the wireless testing and debugging, a sine wave is sent wirelessly, to measure the frequency sample rate through the air (12.5 Hz). If more time permitted, there would be more experimentation on a faster sampling rate through the air to produce more accurate results.

The module that requires the most debugging and testing is the ROM module. It is very tricky to understand how the Flash ROM erases and writes. Also, Nathan Ickes indicated that the display.v, which he created, had problems. Therefore, for several days, it appeared that the Flash ROM was not successfully writing, when it actually had. The display.v was malfunctioning. An additional module was created to read and display the Flash ROM address and data for the debugging process.

Conclusions

The Wireless Musical Project produces satisfying results. The Heart Monitor displays the filtered heart signal. A red, heart-shaped LED formation on the labkit blinks according to the heart rate. The labkit LEDs displays the numerical value of the heart rate. A piece of classical music plays according the heart rate. At a normal heart rate, the classical song plays at normal speed. At a dangerously low heart rate, the song plays much slower but at the same pitch (approximately).

We have learned how to implement each module and interface them together. Not only have we used the knowledge from the class, we have also learned how to use and implement new systems, such as, wireless transceivers, Audio Codec, and Flash ROM. This experience empowers us to design better digital systems in the future.

Acknowledgements

Sinit and Amy would like to thank Jenny Lee, Charlie Kehoe, Chris Forkers, Nathan Ickes, Anantha Chandrakasan, and Keith Kowal for all the help and support. Jenny, the primary TA, provided an incredibly amount of encouragement, sustenance (donuts, coffee, fun Asian snacks), and technical support. Amy would like to thank Charlie Kehoe, Shirley Li, and Nathan Ackerman for their help in the wireless.