DCC – Project proposal
Design of embedded systems, advanced course
EDA385

Jon Axelsson
Jakob Hägg
Per Söderberg

In several power electronic applications it is elementary to control a current through a load. The load is usually inductive and the current through it is large. Hence it is a natural approach to use switching power electronics to keep the losses small. The fact that the load is inductive makes it possible to maintain a dc current by injecting energy in terms of voltage pulses. If this is done in the right way, i.e. the voltage is turned on and off at the right times the result will be a dc current with the wanted mean value and a little ripple.

There are many approaches to achieve this. One way is to use a modulator with a voltage reference as input. The voltage reference is controlled by a PI controller which inputs are the current reference and the actual current . However, the approach in this project is far more rudimentary; if the current is to small compared to the the reference the voltage is turned on. If the current is to big compared to the the reference the voltage is turned off. In practice this will result in far to many switches to be useful, hence a hysteresis band is inserted between the turn on and turn off current.

To make the this controller a little more interesting the size of the hysteresis band could be of variable size. This makes it possible to for example control the switching frequency.

The goal of the project is to make a system to evaluate this type of controller. There will be two user interfaces. The main user interface will be a program running on a PC that is connected to the FPGA. This will be user friendly interface where the user is able to set references and hysteresis for example. This interface will use RS-232 for communication. The other user interface will be based on the switches and leds on the Xlink board and used for less frequently used settings and fault signals.

A block diagram of the system is presented in figure 1. The power electronics, *PE*, and electric motor is already available and the interface is basically a custom designed peripheral unit. A block diagram of the interface is presented in figure 2. The DAC is one of the simplest imaginable, namely a R-2R-ladder witch is very easy to interface since the input is 10 parallel lines. As seen in the diagram the interface to the FPGA is only digital using general input and outputs.

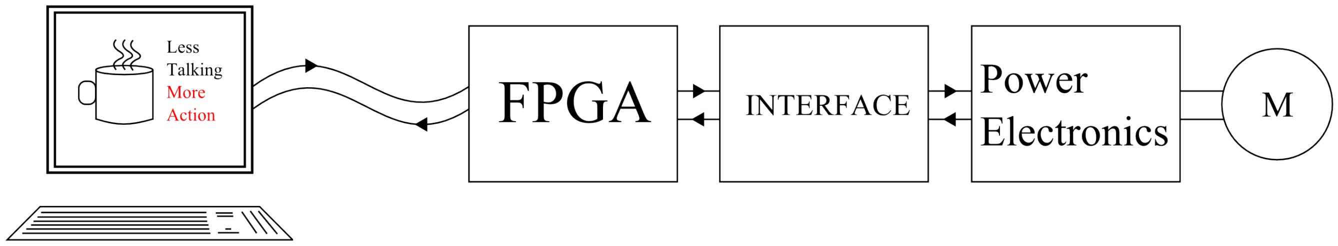

Figure 1: Block diagram of the system.

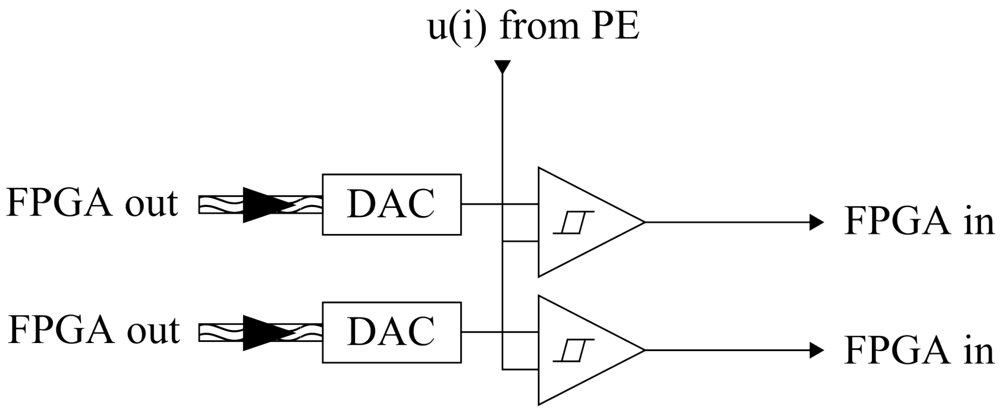

Figure 2: Block diagram of electronic interface.

As seen in figure 3 the components that are needed in the FPGA is a RS-232 controller and a micro processor. The program on the microprocessor will handle the serial communication protocol and the controller. The controller will be implemented in software, this will make it easy to modify the controller from the user interface. It also makes it possible to choose between different controllers. The frequency counter takes the two data lines that controls the power electronics as input and measures the switching frequency. The result is fed back to the controller in the micro processor. The block containing logic is suppose to take the signals shown in figure 2 that goes to the FPGA and convert them into interrupts to the micro processor. A timer will be needed to ensure that the switching does not occur to often, as this could damage the power electronics. The general I/O on the FPGA-board is used for simple test signals and indicating status and errors.

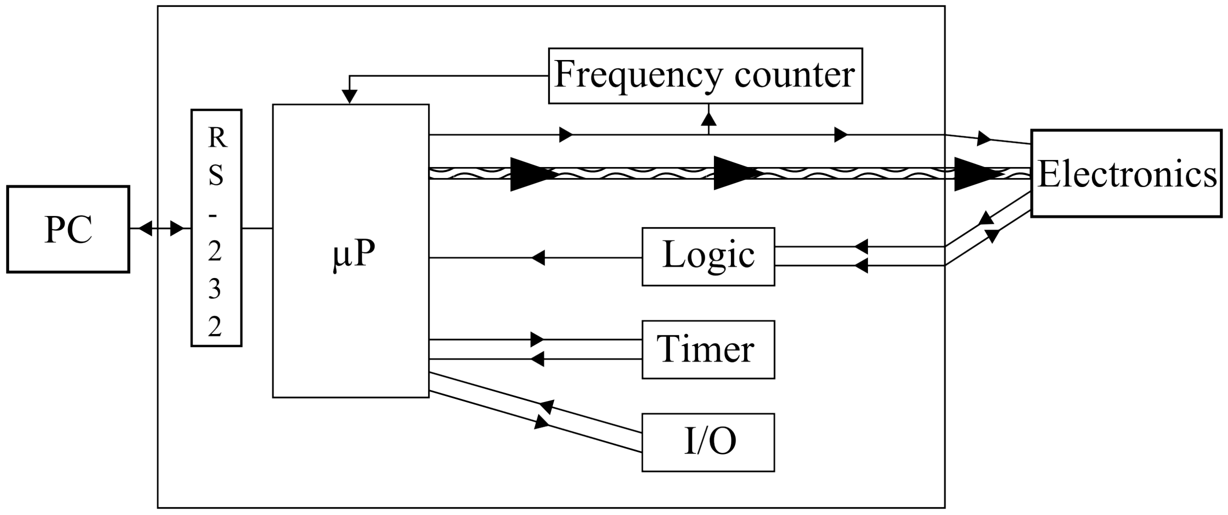

Figure 3: Block diagram of the components in the FPGA.

The protocol for the serial communication will consist of messages. The first byte will contain what type of message it is, for example it could be start/stop-signals, controller parameters or a request for the actual switching frequency. After that comes data if that is needed. Since the system will be in a electrically noisy environment parity check will probably be enabled and a extra byte containing a CRC for the whole message will be implemented.

It is hard to estimate how long time this system will take to implement, test and evaluate, but a solution for a DC-current machine seems very reasonable. Experience tells us that things takes more time than it first seems. If the system works well it would be very interesting to make an implementation for a three-phase permanent magnetized synchronous machine.

Possible expansions for the project except a three-phase version could be a more rigorous user interface with possibilities to test step-response of the controller for example. A DAC with more bits maybe with a some sort of serial interface along with a more advanced controller with more safety precautions is also possible expansions.

Work plan

- Hardware interface
- Protocol to PC
- User interface v 1.0
- Processor, counter, logic, timer
- User interface v 1.1
- Debugging of modules and system
- Get things together and make a dry run
- Test and evaluate DCC
- Three-phase machine ?
- Write report and prepare presentation