Electromechanical escapement system of a pendulum clock using z64 processor

by Antonio Bernardini

1 Project

1.1 Requirements

A z64 processor controls the electromechanical escapement system of a pendulum clock, as per figure. In this system a pair of PHOTOCELLS detects the passage of the pendulum in two very precise positions, interacting asynchronously with the z64 processor.

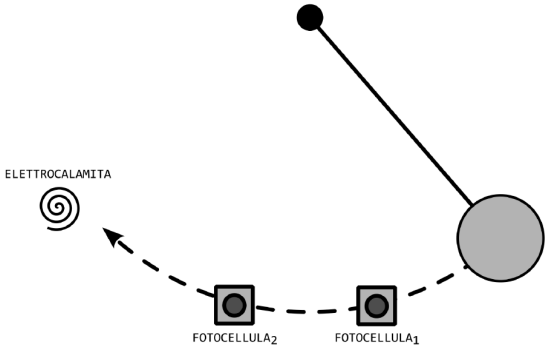

Figure 1. The electromechanical escapement system of a pendulum clock

The time between passing in front of PHOTOCELL$_1$ and PHOTOCELL$_2$ is measured using a TIMER synchronous device. This device, once started, begins to measure time, at the grain of the milliseconds. When the z64 processor subsequently shuts down the device, the elapsed time can be acquired through a specific interface register.

Once the time spent passing in front of the two PHOTOCELLS has been determined, the z64 activates the device ELECTROMAGNET for a time equal to $\frac{1}{64}$ of the passage time, in order to provide additional energy to the pendulum TIMER can also be used to measure this time. Note that since there is only one ELECTROMAGNET, it is not of interest to calculate the time elapsed in the passage between PHOTOCELL$_2$ and PHOTOCELL$_1$.

To design:

- The interfaces of the PHOTOCELL, TIMER, ELECTROMAGNET peripherals
- All the software necessary to operate the system, including any drivers

Tip: starting and stopping TIMER can be interpreted as two different operations.

1.2 Implementation

1.2.1 Hardware

The ELECTROMAGNET peripheral is represented as a simple button on which you can directly write the values 0 and 1 to turn the device on/off:

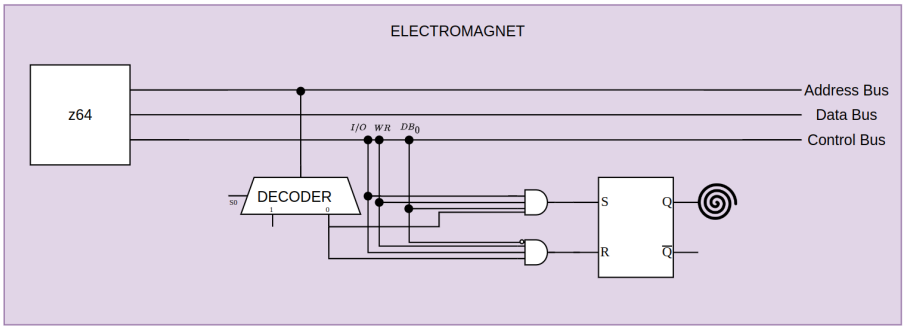

Figure 2. The `ELECTROMAGNET` peripheral

The `PHOTOCELL` peripheral is represented as a device that operates in *mixed mode*, i.e. both *synchronously* and *asynchronously* but ensuring that we have control for interruption requests:

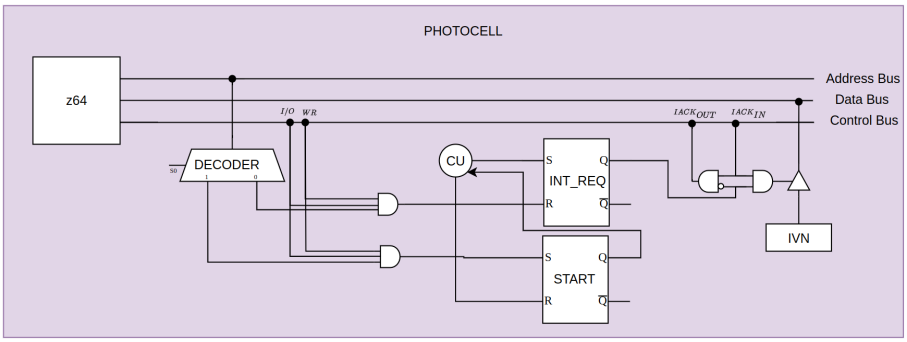

Figure 3. The `PHOTOCELL` peripheral

The `TIMER` peripheral requires to be explicitly started and stopped:

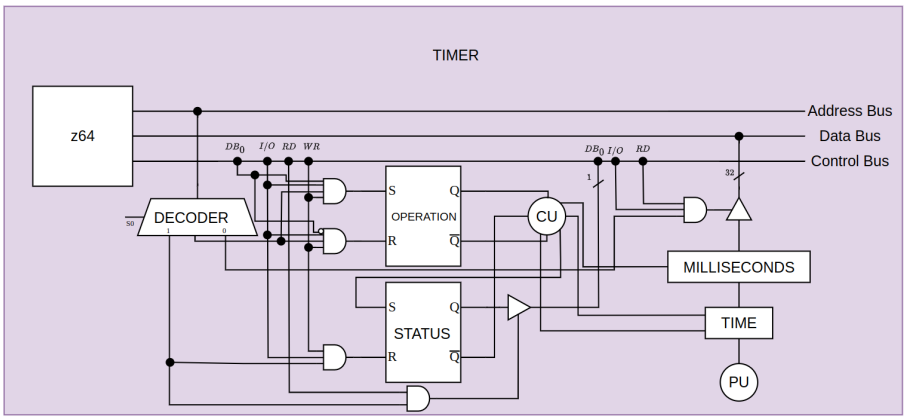

Figure 4. The `TIMER` peripheral

1.2.2 Firmware

So, a possible *firmware implementation* can be found here.